



Universidad Nacional Mayor de San Marcos

Universidad del Perú. Decana de América

Facultad de Ciencias Matemáticas

Escuela Profesional de Matemática

**Métodos de optimización para la gestión de producción
de energía eléctrica bajo incertidumbre**

TESIS

Para optar el Título Profesional de Licenciado en Matemática

AUTOR

Rolando Fredy ICHPAS TAPIA

ASESOR

Erik Alex PAPA QUIROZ

Lima, Perú

2019



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

Referencia bibliográfica

Ichpas, R. (2019). *Métodos de optimización para la gestión de producción de energía eléctrica bajo incertidumbre*. Tesis para optar el título profesional de Licenciado en Matemática. Escuela Profesional de Matemática, Facultad de Ciencias Matemáticas, Universidad Nacional Mayor de San Marcos, Lima, Perú.

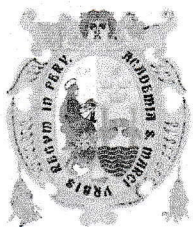
HOJA DE METADATOS COMPLEMENTARIOS

Código Orcid del autor :
Código Orcid del asesor : 0000-0002-8678-6918
DNI del autor : 40708192
Grupo de investigación : No pertenece a ningún grupo de Investigación

Institución que financia parcial o totalmente la investigación : Ninguna

Ubicación geográfica donde se desarrolló la investigación (incluir localidades y coordenadas geográficas) : Calle Los Sauces Mz C Lt 14, San Miguel. (-12.061657, -77.082758)

Año o rango de años que la investigación abarco : Agosto 2018 – Julio 2019



UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS

(Universidad del Perú, DECANA DE AMÉRICA)

FACULTAD DE CIENCIAS MATEMÁTICAS

Ciudad Universitaria - Av. Venezuela S/N cuadra 34

Teléfono: 619-7000. Anexo 1610

Correo Postal: 05-0021. E-mail: eapmat@unmsm.edu.pe

Lima - Perú

Escuela Profesional de Matemática

ACTA DE SUSTENTACIÓN DE TESIS PARA OPTAR EL TÍTULO PROFESIONAL DE LICENCIADO EN MATEMÁTICA

En la UNMSM - Ciudad Universitaria - Facultad de Ciencias Matemáticas, siendo las12:00..... horas del Jueves 5 de setiembre de 2019, se reunieron los docentes designados como Miembros del Jurado Evaluador de Tesis: Mg. Edinson Raúl Montoro Alegre (PRESIDENTE), Mg. María Elizabeth Puelles Bulnes (MIEMBRO), Dr. Erik Alex Papa Quiroz (MIEMBRO ASESOR), para la sustentación de la tesis titulada: «MÉTODOS DE OPTIMIZACIÓN PARA LA GESTIÓN DE PRODUCCIÓN DE ENERGÍA ELÉCTRICA BAJO INCERTIDUMBRE», presentado por el señor Bachiller ROLANDO FREDY ICHPAS TAPIA, para optar el Título Profesional de Licenciado en Matemática.

Luego de la exposición del tesista, el Presidente del Jurado invitó a dar respuestas a las preguntas que le formulen.

Hecha la evaluación correspondiente por los Miembros del Jurado, el tesista mereció la aprobación unánime obteniendo como calificativo promedio la nota de:

.....Dieciocho..... Sobresaliente..... (18).

A continuación, el Presidente del Jurado, Mg. Edinson Raúl Montoro Alegre, manifestó que el señor Bachiller ROLANDO FREDY ICHPAS TAPIA, en vista de haber aprobado la sustentación de su tesis, será propuesto para que se le otorgue el Título Profesional de Licenciado en Matemática.

Siendo las13:00..... horas se levantó la sesión firmando para constancia la presente Acta en tres (3) copias originales.

MG. EDINSON RAÚL MONTORO ALEGRE
PRESIDENTE

MG. MARÍA ELIZABETH PUELLES BULNES
MIEMBRO

DR. ERIK ALEX PAPA QUIROZ
MIEMBRO ASESOR

*“La incertidumbre no está en las cosas, sino en nuestra mente. Esta, es una medida de nuestra
ignorancia...”*

Agradecimientos

En primer lugar, me gustaría agradecer a mi ascesor el Dr. Alex Papa Quiroz por confiarme este proyecto, por la recomendación y gestión para el financiamiento de la tesis y por el apoyo que me ha brindado durante todo el desarrollo de la misma. También me gustaría agradecer a mis padres y hermanos, que con su ejemplo me han enseñado que con esfuerzo siempre es posible alcanzar cualquier objetivo. También agradezco a los miembros del jurado por sus comentarios y observaciones que me ayudaron a mejorar la calidad de esta tesis. Finalmente, agradecer también a todos mis compañeros y amigos cercanos de la universidad, con los cuáles aprendí bastante en esa breve etapa de mi vida académica.

Resumen

En esta tesis, estamos interesados en resolver problemas de optimización en altas dimensiones que surgen en la gestión de la producción de energía eléctrica debido a la aleatoriedad de los datos. Más específicamente, consideramos el problema de la planificación a mediano plazo (MTPP, por sus siglas en inglés) de la producción de energía eléctrica bajo incertidumbre que apunta a la gestión de diversas plantas de producción durante un período de varios meses (incluso años). En general, estos sistemas deben satisfacer una gran variedad de estrictas restricciones operativas en un entorno de toma de decisiones altamente incierto. En este contexto, establecemos los tres acoplamientos naturales de tiempo-espacio-aleatoriedad relacionados con esta aplicación, en donde la aleatoriedad de los datos es modelado por una filtración de un proceso estocástico. Los métodos que recogemos para la discretización del proceso estocástico, es aquel desarrollado en [Lenoir, 2008], los cuáles se basan en una estimación no paramétrica sobre las esperanzas condicionales propuestas en [Nadaraya, 1964; Watson, 1964]. El enfoque que usamos posteriormente para resolver el problema discretizado, consiste en la optimización no diferenciable de la suma de dos funciones convexas, los cuáles son abordados mediante la teoría de los operadores monótonos. Este planteamiento, nos permite estudiar las propiedades de convergencia de diferentes algoritmos clásicos y modernos en un modo unificado como algoritmos de punto fijo. Además, examinamos algunas familias de algoritmos de descomposición-coordinación resultantes de los “*operator splitting methods*”, a saber, los métodos Forward-Backward, Douglas-Rachford y Peaceman-Rachford, los cuáles permiten la implementación de algoritmos proximales paralelos y distribuidos para resolver problemas de optimización convexa. En particular, un algoritmo sobre el cuál estamos fuertemente interesados, es el “*Alternating Direction Method of Multipliers*” (ADMM) que comparte lazos con, por un lado, los métodos Lagrangianos aumentados y, por otro, con la teoría de los operadores *splitting* visto como un caso especial del algoritmo de Douglas-Rachford. Finalmente, implementamos dos modelos matemáticos sencillos (con sus respectivas soluciones numéricas) para la gestión de la producción de energía eléctrica; el primero de naturaleza estocástico y el segundo de naturaleza determinístico.

Palabras clave: Operadores monótonos, optimización convexa, iteración del punto fijo, métodos de división, programación estocástica, algoritmos proximales.

Abstract

This manuscript deals with large-scale optimization problems, arising in the electricity unit commitment problems due to the randomness of the data. More specifically, we consider the mid-term production planning problem (MTPP) of the generation of electrical energy under uncertainty, whose objective is to manage a set of interconnected plants over several months (even years). In general, these systems must satisfy tight operational constraints in an highly uncertain decision environment. In this context, we set the natural time-space-randomness couplings related to this application, where the randomness of the data is modeled by a filtration of a stochastic process. The methods that we collect for the discretization of the stochastic model, are those developed in [Lenoir, 2008], each of them based on non-parametric estimation of conditional expectations proposed in [Nadaraya, 1964; Watson, 1964]. To solve this discretized problem, we use the mathematical model consisting of the sum of two convex functions, which are addressed by the monotonous operators theory. This approach allows us to study the convergence properties of a very wide variety of algorithms, ranging from classical to recently developed, in a uniform way as fixed-point algorithms. In addition, we examine some families of decomposition-coordination algorithms resulting from operator splitting methods, namely, the Forward-Backward, Douglas-Rachford and Peaceman-Rachford splitting methods, which allow the implementation of parallel and distributed proximal algorithms for solving convex optimization problems. In particular, an algorithm on which we are strongly interested is the Alternating Direction Method of Multipliers (ADMM) that shares ties with, on the one hand, augmented Lagrangian methods and, on the other, with splitting operator theory seen as a special case of the Douglas-Rachford algorithm. Finally, we implemented two simple mathematical models (with their respective numerical solutions) for the electricity unit commitment; the first model is stochastic and the second is deterministic.

Keywords: Monotone operators, convex optimization, fixed point iteration, splitting methods, stochastic programming, proximal algorithms.

Notaciones

\mathbb{R}	El conjunto de los números reales.
\mathbb{R}_+	El conjunto $\{\lambda \in \mathbb{R} / \lambda \geq 0\}$.
\mathbb{R}_{++}	El conjunto $\{\lambda \in \mathbb{R} / \lambda > 0\}$.
$\overline{\mathbb{R}}$	El conjunto $\mathbb{R} \cup \{\pm\infty\}$.
\mathbb{R}^n	El conjunto $\{x = (x_1, \dots, x_n) : x_i \in \mathbb{R} \ \forall i = 1, \dots, n.\}$
$2^{\mathbb{R}^n}$	La clase de todos los subconjuntos de \mathbb{R}^n .
$\langle \cdot \cdot \rangle$	El producto escalar (interno) en \mathbb{R}^n .
$\ \cdot \ $	Norma asociada al producto interno en \mathbb{R}^n .
$\Gamma_0(\mathbb{R}^n)$	El conjunto de las funciones $f : \mathbb{R}^n \rightarrow \mathbb{R}$ que son convexas, propias y semicontinuas inferiormente.
$\operatorname{argmin}(f)$	El conjunto $\{x \in \mathbb{R}^n / f(x) = \min f(\mathbb{R}^n)\}$, donde $f \in \Gamma_0(\mathbb{R}^n)$.
$\operatorname{fix}(T)$	El conjunto $\{x \in \mathbb{R}^n / T(x) = x\}$, donde $T : \mathbb{R}^n \rightarrow \mathbb{R}^n$.
$\nabla f(x)$	Gradiente de una función en el punto x .
$\partial f(x)$	Subdiferencial de una función convexa en el punto x .
\mathcal{T}	El operador multivaluado que mapea \mathbb{R}^n en $2^{\mathbb{R}^n}$.
$\operatorname{zer}(\mathcal{T})$	El conjunto $\{z \in \mathbb{R}^n / 0 \in \mathcal{T}(z)\}$.
$\mathcal{R}_{\mathcal{T}}$	Operador resolvente del operador multivaluado \mathcal{T} .
$\mathcal{C}_{\mathcal{T}}$	Operador Cayley del operador multivaluado \mathcal{T} .

Índice

Agradecimientos	V
Resumen	VII
Abstract	IX
Notaciones	XI
1. Introducción	1
1.1. Motivación	1
1.2. Antecedentes del problema	1
1.3. Situación problemática	4
1.4. Objetivos de la investigación	5
1.4.1. Objetivo general	5
1.4.2. Objetivos específicos	5
1.5. Organización y aportes de la tesis	6
I Conceptos Previos	9
2. Estudio de los Operadores Monótonos	11
2.1. Operadores Multivaluados	11
2.2. Operadores No-expansivos y contracciones	12
2.3. Operadores Monótonos	14
2.4. La iteración del Punto Fijo	17
2.4.1. Condiciones para la convergencia del algoritmo de Punto Fijo	18
2.4.2. Método del Gradiente	19
2.4.3. El algoritmo Forward	20
2.5. Construcción del algoritmo del punto proximal generalizado para ∂f	22
2.5.1. Interpretación geométrica del operador Proximal	23
2.5.2. Interpretación geométrica de la envolvente de Moreau-Yosida	24
2.6. Resolvente de Operadores Monótonos	27
3. Operadores Splitting y Algoritmos distribuidos	29
3.1. El algoritmo de división de Peaceman-Rachford	29
3.2. El algoritmo de división de Douglas-Rachford	30

3.3.	El algoritmo de división Forward-Backward	31
3.3.1.	El algoritmo del Gradiente Proximal	32
3.4.	El algoritmo ADMM (Método de dirección alternada de multiplicadores)	34
3.4.1.	Deducción del algoritmo	35
3.4.2.	Análisis de la convergencia	37
3.4.3.	Condiciones para la optimalidad	38
3.4.4.	Criterio de finalización	39
3.5.	Optimización distribuida de consenso con ADMM	40
3.5.1.	Protocolo de consenso con ADMM	41
3.6.	Experimentación numérica	43
II	Modelización y resolución del problema	47
4.	La gestión de la producción de energía eléctrica sujeto a incertidumbres	49
4.1.	Modelización de la gestión de la producción eléctrica	50
4.2.	Modelización de las incertidumbres	52
4.3.	Introducción a la Programación Estocástica	53
4.3.1.	El enfoque “Espera y verás” (Wait and See)	54
4.3.2.	El enfoque “Aquí y ahora” (Here and Now)	55
4.3.3.	El enfoque en “Valor esperado” (Expected value)	55
4.3.4.	Evaluación y comparación de los diferentes enfoques	55
4.4.	Modelo con Recursos	56
4.4.1.	Modelo con recursos en dos etapas	57
4.4.2.	Modelo con recursos multietapas	58
5.	Aplicación: La gestión de la producción de energía eléctrica	65
5.1.	Descripción de las diferentes tecnologías de producción de E.E. en el Perú	65
5.1.1.	Centrales Termoeléctricas	66
5.1.2.	Centrales Hidroeléctricas	66
5.1.3.	Centrales de Generación con Recursos Energéticos Renovables (RER)	68
5.2.	Un primer modelo para la gestión de la producción de E.E.	68
5.2.1.	Solución numérica	71
5.3.	Un segundo modelo para la gestión de la producción de E.E.	72
5.3.1.	Hipótesis	73
5.3.2.	Construcción del algoritmo	73
5.3.3.	Solución numérica	74
5.4.	Conclusiones	79
	Apéndice	81
A.	Revisión del análisis convexo	83
A.0.1.	Conjuntos convexos	85
A.0.2.	Funciones convexas	88
B.	Introducción a los Procesos Estocásticos	93
B.0.1.	Construcción Formal de la Estructura Probabilística de un Proceso Estocástico	96
B.0.2.	Filtración	98

C. Lista de algoritmos en Matlab	99
C.1. Algoritmos del Capítulo 3	100
C.2. Algoritmos del Capítulo 5	107
C.2.1. Algoritmos del Primer modelo	107
C.2.2. Algoritmos del Segundo modelo	115
Bibliografía	123

Capítulo 1

Introducción

1.1. Motivación

Este trabajo de investigación titulado “*Métodos de optimización para la Gestión de la producción de Energía Eléctrica bajo incertidumbre*”, ha sido desarrollado con la motivación de introducir un nuevo modelo matemático para la toma de decisiones en la gestión de la producción de la energía eléctrica en el Perú, teniendo en cuenta las incertidumbres que puedan surgir por efectos de la volatilidad del clima, el mercado o el consumo de energía; e incluso por la incorporación de nuevas fuentes de energía renovables tales como la energía eólica, solar, entre otras; el cuál, tiene como fin fundamental el beneficio del usuario y la eficiencia del sector eléctrico peruano.

1.2. Antecedentes del problema

El sistema de energía eléctrica, es uno de los sistemas de servicio más importantes que mantiene a nuestra sociedad en funcionamiento, ya que es responsable de generar, transmitir y distribuir electricidad, que alimenta casi todos los aspectos de nuestra vida. Estos tres componentes principales del sistema de energía eléctrica: generación, transmisión y distribución; constituyen una red multinivel que conecta los suministros de energía iniciales con los usuarios finales para los usos diarios que estos demandan. En el Perú, el sistema eléctrico de transmisión está conformado por el SEIN, y por las Redes de los Sistemas Aislados (SSAA). El sistema de distribución, encargada de transmitir grandes volúmenes de energía hasta los usuarios finales, es tratado de forma independiente. La potencia eléctrica, cuya unidad de medida es el watt (W), cuantifica la cantidad de energía que se consume, produce o traslada en cada unidad de tiempo; mientras que la energía eléctrica representa la cantidad total de energía que se consumió, produjo o trasladó durante un determinado periodo, por lo que su unidad de medida suele ser el watt-hora (Wh). Por ejemplo, si la potencia de una lámpara eléctrica es 100 W y esta permanece encendida por dos horas, entonces, la energía eléctrica consumida sería 200 Wh.

En los últimos 20 años se ha observado que, en paralelo al crecimiento de la economía mundial, la demanda de energía eléctrica se ha incrementado de manera sostenida. De esta manera, se pasó de consumir 11 260 TWh en 1990 a 22 662 TWh en 2015 (aumento de casi 100 %); mientras que en el Perú la demanda creció en más de 200 % al pasar de 10.7 TWh en 1992 a 42.3 TWh en 2015 (ver Fig 1.1), con una tasa de crecimiento promedio anual de aproximadamente 5.1 %.

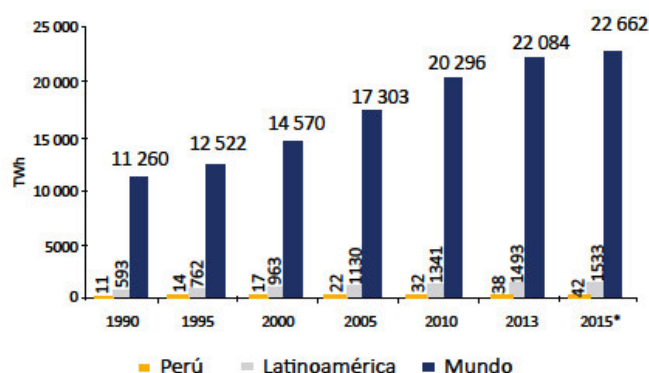


Figura 1.1: Evolución de la demanda de electricidad en el Perú, Latinoamérica y el mundo

* Fuentes: ONU, Enerdata y MEM. Elaboración: GPAE-Osinergmin.

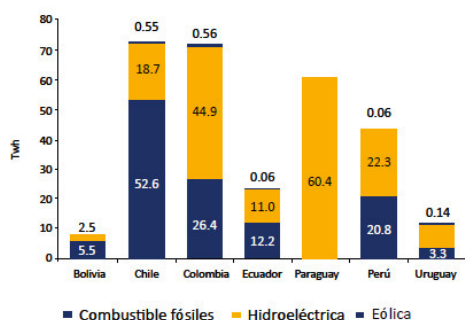


Figura 1.2: Generación de electricidad por fuente, 2013 (Resto de Sudamérica)

* Fuente: ONU. Elaboración: GPAE-Osinergmin.

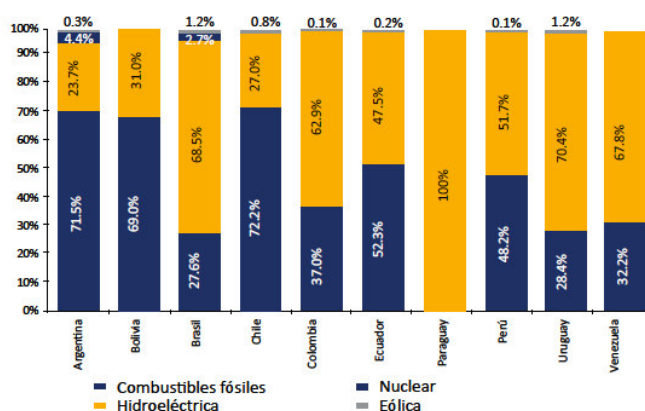


Figura 1.3: Participación de la generación electricidad por fuente, 2013

* Fuente: ONU. Elaboración: GPAE-Osinergmin.

En todo el mundo, las principales fuentes utilizadas para generar energía eléctrica siguen siendo los combustibles fósiles, como el carbón, el gas natural, el petróleo, la energía nuclear, etc., mientras que las fuentes renovables (por ejemplo, la eólica, la solar, etc.) están ganando una mayor participación recientemente. La producción por fuentes de energía en Sudamérica es, principalmente, de origen hidroeléctrico y térmico. Su composición es diversa, dependiendo de cada país, y es la región con mayor participación renovable relativa en su correspondiente matriz eléctrica. Así, a 2013, Paraguay produjo 100 % de energía hidroeléctrica, Brasil 68.5 % y Perú 51.7 %. La participación de cada fuente de energía en cada país se presenta en los gráficos 1.2 y 1.3.

El 1 de mayo de 2008 es una fecha importante para el proceso de transformación del modelo energético de Perú, ya que ese día se aprobó y promulgó por la Presidencia de la República, el Decreto Legislativo 1002 que establece las bases de la promoción de la inversión en sistemas renovables de generación eléctrica [Congreso-Rep-Perú, 2008]. El 23 de marzo de 2011, el MINEM, actualizó mediante Decreto Supremo N° 012-2011-EM el Reglamento de Generación de Electricidad con Energías Renovables [MINEM, 2011]. A continuación se presentan un breve comentario respecto al artículo 2 del mencionado Decreto Legislativo 1002. *“El Artículo 2 declara de interés nacional y necesidad pública la promoción y participación de la energía procedente de Recursos*

Energéticos Renovables (RER), en la matriz energética nacional. Asimismo, indica que el MINEM deberá establecer un porcentaje objetivo de participación de los sistemas RER en la matriz energética nacional. Para el primer quinquenio se estableció que el mencionado porcentaje objetivo sea igual al 5 % del consumo nacional de electricidad”.

La implementación de este D.L. plantea dos aspectos fundamentales que en este punto vale la pena señalar:

- (1) Los sistemas (RER) se caracterizan por su variabilidad en su costo de producción. A manera de ejemplo, en la figura 1.4 se muestran los costos totales de tres tecnologías de generación eléctrica (hidroeléctrica, central térmica a gas natural y central térmica a diesel) según el número de horas de funcionamiento a lo largo de un año.
- (2) Existen diversos factores que afectan a los mercados de electricidad, los cuáles tienen efectos sobre la infraestructura eléctrica, la operación del sistema y las tarifas de electricidad. Como se señala en [Osinergmin, 2016], de manera general estos factores se pueden agrupar en tres tipos: climáticos geológicos, económicos y sociales:
 - **Factores climáticos y geológicos:** Las centrales de generación eléctrica necesitan de insumos para su funcionamiento. Algunos de ellos se obtienen de forma natural en el ambiente, por ejemplo los recursos hídricos, la energía del sol o del viento. Otros se pueden obtener vía transacciones en el mercado, como los combustibles fósiles (productos derivados del petróleo, gas natural y carbón). Los factores climáticos y geológicos afectan la disponibilidad de estos insumos para la generación eléctrica, así como a la infraestructura de la industria eléctrica. Entre estos factores se pueden mencionar fenómenos climáticos como huracanes y tifones, y geológicos, como movimientos sísmicos de gran magnitud. Estos últimos pueden ocasionar destrucción de centrales de generación y la caída de torres de transmisión que podrían dejar sin suministro eléctrico incluso a grandes ciudades. En el caso de eventos climáticos, por ejemplo, la industria eléctrica de Sudamérica es particularmente sensible a la ocurrencia del fenómeno de El Niño, que llega a afectar el desarrollo normal del mercado desde el nivel de generación, transmisión y distribución, debido a cambios en las precipitaciones, temperatura, patrones de viento, radiación solar, entre otros.
 - **Factores económicos:** Entre los factores económicos que afectan el mercado eléctrico se encuentran aquellos relacionados a variables que tienen impacto sobre las tarifas de electricidad. Por ejemplo, los precios de los insumos utilizados por las centrales de generación como el diésel, el gas natural, el carbón, etc. No obstante, es conveniente precisar que el efecto de variaciones en los precios dependerá, en gran medida, de la configuración del parque de generación eléctrico. De esta manera, países que basan su producción en centrales térmicas que utilizan derivados del petróleo y que a su vez no cuentan con reservas propias de estos productos, se verán más afectados por cambios en los precios internacionales de estos insumos, mientras que aquellos países cuya generación eléctrica es realizada en una mayor proporción por centrales hidroeléctricas no se verían tan afectados. En el caso del Perú, otros factores económicos que tienen incidencia sobre las tarifas eléctricas son aquellos relacionados a variables macroeconómicas que reflejan la situación del mercado internacional como, por ejemplo, la inflación internacional medida por diferentes tipos de índices (por ejemplo, el índice de precios al productor-IPP y el

índice de precios al consumidor-IPC); precios de metales empleados en la construcción de redes eléctricas como el aluminio y el cobre; y el tipo de cambio, etc.

- **Factores sociales:** Dentro de los factores sociales se encuentran las protestas de la población debido a la construcción de centrales de generación que podrían afectar el ambiente. Por ejemplo, en el caso de las centrales hidroeléctricas, los pobladores que residen en el área de influencia podrían no estar de acuerdo debido a que perciben que la construcción afectará sus actividades económicas. También se encuentran dentro de estos factores el rechazo a un tipo particular de tecnología. Un ejemplo es la generación en base a energía nuclear, que en algunos países como Alemania, ha llevado a establecer compromisos sobre el cierre de sus centrales nucleares debido al alto riesgo que representa sobre la salud humana. Los temores se han acrecentado a partir del desastre de Fukushima, entre otros.

Sobre la base de estas dos observaciones, así como el crecimiento acelerado de la demanda energética; resulta necesario la implementación de nuevas técnicas matemáticas que permitan interconectar y gestionar de manera eficiente la producción de las diferentes fuentes de energía eléctrica y que a la vez, estos modelos se adapten mejor a las variaciones intrínsecas producidas por el entorno aleatorio. En este proyecto, el nuevo modelo de optimización para la gestión de la producción de la energía eléctrica bajo incertidumbre que proponemos, permitiría satisfacer la demanda energética y a la vez protegerlos de la volatilidad del mercado, las variaciones climáticas y otros posibles escenarios cambiantes del entorno de manera óptima.

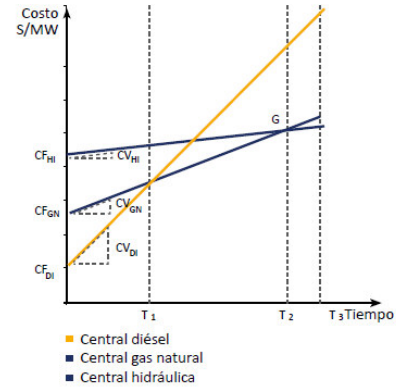


Figura 1.4: CF (Costo fijo), CV (Costo variable).

* Fuente: Oren (2000). Elaboración: GPAE-Osinergmin.

1.3. Situación problemática

La producción de energía eléctrica, debe gestionarse teniendo en cuenta las incertidumbres con el fin de garantizar, a un costo menor el equilibrio entre la oferta y la demanda en el futuro, pero también, teniendo en cuenta las necesidades inmediatas. Por lo tanto, son necesarias varias visiones del problema, los cuáles deben tomar en cuenta el horizonte temporal, y cuanto más la visión abarca un largo período de tiempo, la descripción de los peligros jugará un papel cada vez más importante. En [Wallace & Fleten, 2003] y [Conejo *et al.*, 2010] se describen algunas de estas diferentes perspectivas que mencionaremos brevemente a continuación:

- **Problema de Planificación a Largo Plazo (PPLT):** Dado que la demanda de electricidad cambia con el tiempo, la estructura del parque de generación eléctrica debe adaptarse y desarrollarse. Por lo tanto, se trata de tomar decisiones de inversión, es decir, de construir nuevas plantas. El horizonte temporal comienza dos o tres años después del momento presente (que corresponde al tiempo de construcción de una planta), y se extiende hasta 25 años después, lo que permite tener en cuenta la duración del funcionamiento de las centrales. Es necesario decidir entre

plantas que requieren costos de construcción muy altos (y desmantelamiento) pero que aseguran un costo de producción bajo, como la tercera generación nuclear, y aquellas con características opuestas, como los ciclos combinados de gas. Para modelar los peligros que pueden influir en las decisiones, cada año se construyen cuatro escenarios bastante diferentes para representar las diversas posibilidades para cambiar las condiciones económicas.

· ***Problema de Planificación a Mediano Plazo (PPMT)***: Se trata de anticipar las decisiones óptimas para garantizar el equilibrio entre la oferta y la demanda a un costo menor, en un horizonte superior al mes actual y que generalmente abarca varios meses o incluso algunos años (algunos autores consideran hasta 5 años). Más específicamente, es necesario establecer las paradas de las unidades energéticas, desarrollar estrategias de gestión de inventario, medir los riesgos físicos y financieros y definir las acciones de cobertura en consecuencia.

En esta tesis, nosotros nos focalizamos solo en el PPMT. En este contexto, las dos fuentes principales de complejidad son atribuibles a dos aspectos fundamentales:

1.- *El problema de la dimensionalidad*: Los parámetros aleatorios que resultan de un PPMT, generalmente evolucionan en un proceso estocástico multidimensional y se caracterizan por poseer una distribución de probabilidad continua (y conjunta), cuya forma discretizada genera la aparición de una gran cantidad de variables adicionales, el cuál es un problema de optimización difícil de abordar numéricamente por las técnicas habituales.

2.- *El problema de la no diferenciabilidad*: La aparición de funciones no diferenciables productos del modelado del problema. Esto es, el sistema que se pretendería modelar (el parque de producción eléctrica en este caso) esta conformado por una cantidad significativa de plantas de producción eléctrica con comportamientos muy heterogéneos, cuya interconexión debe ser modelado cuidadosamente.

Por lo tanto, es en estas dos problemáticas sobre el cuál se desarrolla este trabajo.

1.4. Objetivos de la investigación

1.4.1. Objetivo general

Implementar un modelo de optimización estocástica junto con su solución numérica que resuelva de manera eficiente el problema de la gestión óptima de la producción de la energía eléctrica en entornos aleatorios, limitandonos a los métodos que son aplicables a los sistemas en alta dimensión derivados de la discretización del problema estocástico original; para tales sistemas, cada vector de minimización puede contener cientos o incluso miles de variables de decisión en un período arbitrario en donde el horizonte de planificación que contemplamos es a mediano plazo.

1.4.2. Objetivos específicos

- (1) Introducir la teoría de los operadores monótonos y mostrar su conexión con la optimización convexa.
- (2) Establecer la condiciones generales para la convergencia de diferentes algoritmos en un enfoque unificado como derivados del algoritmo del punto fijo.

- (3) Implementar algoritmos proximales paralelos y distribuidos para resolver problemas de optimización en altas dimensiones.
- (4) Evidenciar los tres acoplamientos naturales que surgen en la gestión de la producción de energía eléctrica en entornos aleatorios.
- (5) Estudiar la estructura del árbol de escenarios para modelar las incertezas de los programas estocásticos con recursos multi-etapas (“*Multistage Stochastic Programs with Recourse*”).

1.5. Organización y aportes de la tesis

Este trabajo esta dividido en dos partes. La primera (“*Conceptos Previos*”), involucra dos capítulos en donde se desarrollan los conceptos fundamentales que serán necesarios para implementar las soluciones numéricas a los problemas que se presentarán en la segunda mitad. En consecuencia, en el capítulo 2 abordamos el problema de la no diferenciabilidad. Comenzamos desarrollando la teoría de los operadores monótonos y mostramos su interconexión con el análisis convexo y la teoría de los operadores no-expansivos y, a la vez, estudiamos diferentes algoritmos clásicos de la optimización convexa como casos particulares del algoritmo del punto fijo. Si bien el análisis de toda esta teoría puede desarrollarse en el contexto de los espacios de Hilbert, nosotros nos limitaremos solo al espacio euclideo \mathbb{R}^n ya que este es suficiente para abordar el problema de optimización resultante de nuestro modelo.

En el capítulo 3 estudiamos a los algoritmos de división de Douglas-Rachford, Peaceman-Rachford, Forward-Backward y al ADMM (*Método de dirección alterna de multiplicadores*); junto con sus respectivas condiciones para la convergencia. Estos algoritmos nos permitirán abordar el problema de la dimensionalidad, pues estos toman la forma de un procedimiento de descomposición-coordinación, en el que las soluciones para pequeños subproblemas locales se coordinan para encontrar una solución de un “*gran*” problema global, permitiendo de este modo la implemetación de algoritmos proximales paralelos y distribuidos.

La segunda parte (“*Modelización y resolución del problema*”), viene conformada por 2 capítulos. En el capítulo 4, introducimos el modelo de la gestión de la producción de energía eléctrica en el contexto de los “*grandes sistemas*”. También presentamos los modelos con recursos multietapas, los cuáles son procesos estocásticos que nos permitirán representar las incertezas originados por la acumulación de la información. La discretización de este proceso, se basa en las investigaciones realizadas por *Arnaud Lenoir* [Lenoir, 2008] en el Laboratorio de Informática, de Modelización y Optimización de Sistemas (LIMOS) en Clermont-Ferrand en asociación con el Departamento de Optimización, Simulación, Riesgo y Estadísticas (OSIRIS) de la división de Investigación y Desarrollo de la EDF (Francia)¹.

Finalmente, en el capítulo 5 estudiamos de manera breve las diferentes tecnologías de generación eléctrica en el Perú e implementamos dos simples modelos matemáticos junto con su solución numérica para la gestión de la producción de energía eléctrica considerando tres teconologías: la termoeléctrica, la hidroeléctrica y una arbitraria.

¹El principal productor de electricidad en Europa, el Grupo EDF es una de las compañías de energía líderes en el mundo.

Respecto al aporte de esta tesis, la principal contribución se encuentra en dos aspectos: en el desarrollo didáctico de la teoría y en la implementación computacional junto con los experimentos numéricos realizados de algunos algoritmos modernos que pueden aplicarse a problemas específicos de nuestra realidad.

Parte I

Conceptos Previos

Capítulo 2

Estudio de los Operadores Monótonos

El objetivo en este capítulo es minimizar $f \in \Gamma_0(\mathbb{R}^n)$, es decir:

$$\underset{x \in \mathbb{R}^n}{\text{Minimizar}} \quad f(x),$$

donde en general f es no diferenciable. Para esto, revisaremos de manera breve algunos conceptos previos que serán de bastante utilidad en las 4 primeras secciones y retomaremos nuestro objetivo principal de este capítulo en la sección 2.5.

2.1. Operadores Multivaluados

Definición 2.1. Sean \mathbb{X} e \mathbb{Y} dos conjuntos arbitrarios no nulos sin ninguna estructura en particular, llamamos **operador multivaluado** o simplemente **relación** a toda aplicación del tipo

$$\mathcal{T} : \mathbb{X} \rightarrow 2^{\mathbb{Y}},$$

el cuál también es denotado por

$$\mathcal{T} : \mathbb{X} \rightrightarrows \mathbb{Y}$$

A partir de la definición 2.1 se desprenden las siguientes observaciones:

- \mathcal{T} mapea cada punto $x \in \mathbb{X}$ en un conjunto $\mathcal{T}(x) \subseteq \mathbb{Y}$, es decir; \mathcal{T} es un operador multivaluado, de este modo \mathcal{T} puede ser caracterizado por su *gráfico*:

$$\text{graf}(\mathcal{T}) = \{(x, y) \in \mathbb{X} \times \mathbb{Y} | y \in \mathcal{T}(x)\}. \quad (2.1)$$

- Si Ψ es un subconjunto de \mathbb{X} , entonces

$$\mathcal{T}(\Psi) = \bigcup_{x \in \Psi} \mathcal{T}(x)$$

- El *dominio* y el *rango* de \mathcal{T} estan dados por

$$\text{dom}(\mathcal{T}) = \{x \in \mathbb{X} | \mathcal{T}(x) \neq \emptyset\} \quad \text{y} \quad \text{ran}(\mathcal{T}) = \mathcal{T}(\mathbb{X}), \quad (2.2)$$

respectivamente.

- Sea $\mathcal{R} : \mathbb{Y} \rightrightarrows \mathbb{Z}$, el operador *Composición* $\mathcal{R} \circ \mathcal{T}$ esta definido por

$$\mathcal{R} \circ \mathcal{T} : \mathbb{X} \rightrightarrows \mathbb{Z} : x \mapsto \mathcal{R}(\mathcal{T}(x)) = \bigcup_{y \in \mathcal{T}(x)} \mathcal{R}(y). \quad (2.3)$$

- El operador *Inversa* de \mathcal{T} , denotado por \mathcal{T}^{-1} (el cuál siempre existe) esta definido por

$$\mathcal{T}^{-1}(y) = \{x \in \mathbb{X} | y \in \mathcal{T}(x)\}, \quad (2.4)$$

en términos de gráficos

$$\text{graf}(\mathcal{T}^{-1}) = \{(y, x) \in \mathbb{Y} \times \mathbb{X} | (x, y) \in \text{graf}(\mathcal{T})\}. \quad (2.5)$$

Otra observación importante que vale la pena resaltar es que si $\text{dom}(\mathcal{T}) \neq \emptyset$ y si para cada $x \in \text{dom}(\mathcal{T})$ se tiene que $\mathcal{T}(x) = \{y\}$ es un “*singleton*”, entonces decimos que \mathcal{T} es un *operador* en el sentido clásico, por lo tanto, este puede ser identificado con el operador $T : \text{dom}(\mathcal{T}) \rightarrow \mathbb{Y}$ y lo denotaremos en la manera habitual, es decir $\mathcal{T}(x) = T(x) = y$ en lugar de $\mathcal{T}(x) = \{y\}$ (Observe el cambio del tipo de letra de \mathcal{T} por T) para todo $x \in \text{dom}(\mathcal{T})$. Inversamente, si \mathbb{D} es un subconjunto no vacío de \mathbb{X} , un operador $T : \mathbb{D} \rightarrow \mathbb{Y}$ puede ser identificado con un operador $\mathcal{T} : \mathbb{X} \rightrightarrows \mathbb{Y}$ de la forma:

$$\mathcal{T} : \mathbb{X} \rightrightarrows \mathbb{Y} : x \mapsto \begin{cases} \{T(x)\}, & \text{si } x \in \mathbb{D} \\ \emptyset & \text{si } x \notin \mathbb{D} \end{cases}$$

De este modo se observa que el concepto de *Operador Multivaluado* generaliza de manera natural el concepto de *operador* en el sentido clásico y por ende, también el concepto de *función*. En lo que sigue de la tesis usaremos las notaciones \mathcal{T} y T , en el primer caso para referirnos a operadores multivaluados en general y en el segundo para operadores en el sentido clásico.

Definición 2.2 (Operador). *Un **operador** en \mathbb{R}^n es cualquier operador Multivaluado (incluyendo a los operadores en el sentido habitual) que mapea $\mathbb{R}^n \rightrightarrows \mathbb{R}^n$.*

2.2. Operadores No-expansivos y contracciones

Decimos que el operador multivaluado $\mathcal{T} : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ es Lipschitziano si:

$$\exists L \in \mathbb{R}_+ \quad / \quad \|y - y^*\| \leq L \|x - x^*\|$$

para todo $x, x^* \in \mathbb{R}^n$, $y \in \mathcal{T}(x)$, $y^* \in \mathcal{T}(x)^*$. Esto implica que \mathcal{T} es un operador en el sentido clásico T , ya que si $x = x^*$ tendríamos necesariamente que $y = y^*$. Esto motiva las siguientes definiciones:

Definición 2.3. *Decimos que un operador T es una **contracción** si este es un operador Lipschitziano con constante de Lipschitz $L < 1$.*

Definición 2.4. *Decimos que un operador T es **no-expansivo** si este es un operador Lipschitziano con constante de Lipschitz $L = 1$.*

Definición 2.5. Sea D un subconjunto no vacío de \mathbb{R}^n y sea $T : D \rightarrow \mathbb{R}^n$. Decimos que T es **firmemente no expansivo** si

$$(\forall x \in D)(\forall y \in D) \quad \|Tx - Ty\|^2 + \|(I - T)x - (I - T)y\|^2 \leq \|x - y\|^2; \quad (2.6)$$

La interpretación geométrica de las dos primeras definiciones es bastante simple, el mapeo de un par de puntos por una contracción reduce la distancia entre dichos puntos; pero mapearlos mediante un operador no-expansivo no aumenta la distancia entre ellos.

Notemos que si los operadores T_1 y T_2 tienen constantes de Lipschitz L_1 y L_2 respectivamente, entonces la composición $T_1 \circ T_2$ tiene constante de Lipschitz $L_1 L_2$. Así, la composición de operadores no-expansivos es nuevamente no-expansivo; más aún, la composición de un operador no-expansivo y una contracción es una contracción.

Análogamente, si los operadores T_1 y T_2 tienen constante de Lipschitz L_1 y L_2 respectivamente y si $\alpha_1, \alpha_2 \in \mathbb{R}$, entonces $\alpha_1 T_1 + \alpha_2 T_2$ tiene constante de Lipschitz $|\alpha_1|L_1 + |\alpha_2|L_2$. Así, el promedio ponderado de operadores no-expansivos, i.e., $\theta T_1 + (1 - \theta)T_2$ con $\theta \in [0, 1]$, es también un operador no-expansivo. Si además, uno de ellos es una contracción y $\theta \in (0, 1)$ entonces el promedio ponderado es una contracción.

Podemos definir formalmente la idea del *operador promedio ponderado* de operadores no-expansivos del siguiente modo:

Definición 2.6. Sea D un subconjunto no vacío de \mathbb{R}^n , $T : D \rightarrow \mathbb{R}^n$ un operador y $\alpha \in (0, 1)$. Entonces decimos que T es un **operador ponderado** con constante α , o **α -averaged** si existe un operador no-expansivo $R : D \rightarrow \mathbb{R}^n$ tal que $T = (1 - \alpha)I + \alpha R$.

Respecto a esta definición, existen varias observaciones que se desprenden directamente de las proposiciones y propiedades antes mencionadas, los cuáles serán muy importantes recordar más adelante cuando analicemos la convergencia de diferentes algoritmos.

- Si T es ponderado entonces T es no-expansivo.
- Si T es no-expansivo, este no necesariamente es ponderado, un ejemplo clásico es $T = -I : \mathbb{R}^n \rightarrow \mathbb{R}^n$ cuando $\mathbb{R}^n \neq \{0\}$.
- T es firmemente no-expansivo si y solo si este es $(1/2)$ -averaged.
- T y R tienen los mismos puntos fijos.

Por último, la definición que viene a continuación también tiene una estrecha relación con la no-expansividad.

Definición 2.7 (Operador cocoercivo). Sea D un subconjunto no vacío de \mathbb{R}^n , $T : D \rightarrow \mathbb{R}^n$ un operador y $\beta \in \mathbb{R}_{++}$. Entonces, decimos que T es **β -cocoercivo** si βT es firmemente no expansiva. i.e.,

$$(\forall x \in D)(\forall y \in D) \quad \langle x - y | Tx - Ty \rangle \geq \beta \|Tx - Ty\|^2. \quad (2.7)$$

Proposición 2.1. Sea D un subconjunto no vacío de \mathbb{R}^n y $T : D \rightarrow \mathbb{R}^n$. Entonces T es no-expansivo si y solo si $I - T$ es $(1/2)$ -cocoercivo.

Demostración. Ver [Heinz H. Bauschke, 2017]. □

2.3. Operadores Monótonos

Definición 2.8 (Operador monótono). *Un operador $\mathcal{T} : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ es **monótono** si:*

$$\langle x - x^* | y - y^* \rangle \geq 0 \quad (2.8)$$

para todo $x, x^* \in \mathbb{R}^n$, $y \in \mathcal{T}(x)$, $y^* \in \mathcal{T}(x^*)$.

Definición 2.9 (Operador estrictamente monótono). *Un operador $\mathcal{T} : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ es **estrictamente monótono** si:*

$$\langle x - x^* | y - y^* \rangle > 0 \quad (2.9)$$

para todo $x, x^* \in \mathbb{R}^n$, $y \in \mathcal{T}(x)$, $y^* \in \mathcal{T}(x^*)$.

Definición 2.10 (Operador fuertemente monótono). *Un operador $\mathcal{T} : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ es **fuertemente monótono** de constante $m \in \mathbb{R}_{++}$ si:*

$$\langle x - x^* | y - y^* \rangle \geq m \|x - x^*\|^2 \quad (2.10)$$

para todo $x, x^* \in \mathbb{R}^n$, $y \in \mathcal{T}(x)$, $y^* \in \mathcal{T}(x^*)$.

Uno de los conjuntos clave que nos servirá para analizar la existencia de los mínimos de las funciones en $\Gamma_0(\mathbb{R}^n)$ es el conjunto $\text{zer}(\mathcal{T})$ definido a continuación.

Definición 2.11. *Un zero de un operador monótono $\mathcal{T} : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ es cada punto $z \in \mathbb{R}^n$ tal que $0 \in \mathcal{T}(z)$. El conjunto de todos los zeros es por lo tanto representado por:*

$$\text{zer}(\mathcal{T}) = \mathcal{T}^{-1}(0).$$

Definición 2.12. *Sea $\mathcal{T} : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ un operador monótono. Decimos que \mathcal{T} es **monótono maximal** si su gráfico no está estrictamente contenido en el gráfico de ningún otro operador monótono, i.e., para cada $(x, u) \in \mathbb{R}^n \times \mathbb{R}^n$,*

$$(x, u) \in \text{gra}(\mathcal{T}) \Leftrightarrow \forall (y, v) \in \text{gra}(\mathcal{T}) : \langle x - y | u - v \rangle \geq 0.$$

Teorema 2.1. *Sea $\mathcal{T} : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ monótono. Entonces, existe una extensión monótona maximal de \mathcal{T} , i.e., un operador monótono maximal $\tilde{\mathcal{T}} : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ tal que $\text{gra}(\mathcal{T}) \subseteq \text{gra}(\tilde{\mathcal{T}})$.*

Demostración. Ver [Heinz H. Bauschke, 2017]. □

La gráfica 2.1 muestra la extensión de un operador monótono $\mathcal{T} : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ cuando $n = 1$. Se puede observar claramente que para cualquier operador monótono siempre existirá al menos una extensión maximal, más aún; puede suceder que exista infinitas extensiones maximales, por lo tanto, la elección de una de tales extensiones dependerá exclusivamente del problema que estemos analizando.

Un resultado fundamental de la teoría de los operadores monótonos maximales es el famoso teorema de Minty, el cuál caracteriza de manera elegante la monotonidad maximal.

Teorema 2.2 (Minty). *Sea $\mathcal{T} : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ monótona. Entonces \mathcal{T} es monótona maximal si y solo si $\text{ran}(\mathcal{I} + \mathcal{T}) = \mathbb{R}^n$.*

Demostración. Ver [Heinz H. Bauschke, 2017] □

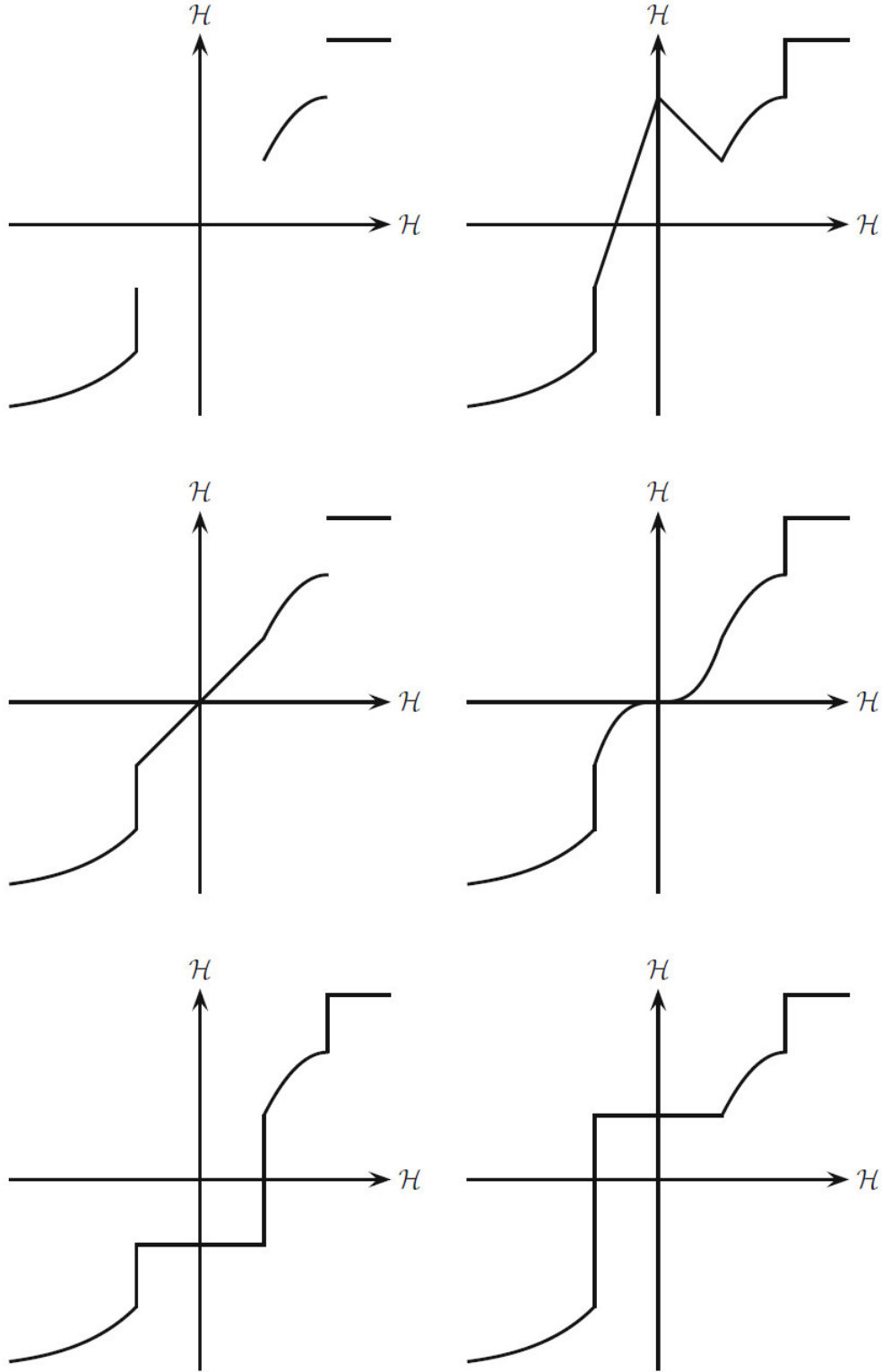


Figura 2.1: *Extensión de un operador monótono $\mathcal{T} : \mathbb{R}^n \rightrightarrows \mathbb{R}^n (n = 1)$. Superior izquierda: Gráfico de \mathcal{T} . Superior derecha: Gráfico de una extensión no monótona de \mathcal{T} . Centro izquierda: Gráfico de una extensión monótona de \mathcal{T} que no es monótona maximal. Centro derecha, inferior izquierda e inferior derecha: Gráfico de una extensión monótona maximal de \mathcal{T} .*

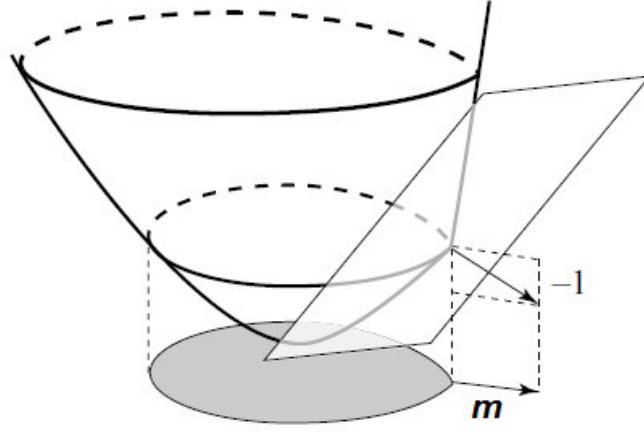


Figura 2.2: Interpretación geométrica del subgradiente.

De entre todos los operadores monótonos maximales, existe una que es fundamental en el estudio de las funciones no diferenciables: esta es la *subdiferencial* de una función en $\Gamma_0(\mathbb{R}^n)$ el cuál, se define como sigue.

Definición 2.13 (Subgradiente, Subdiferencial). Sea $f \in \Gamma_0(\mathbb{R}^n)$. Entonces,

- llamamos “**subgradiente**” de f en el punto x a todo vector $m \in \mathbb{R}^n$ que verifica

$$f(y) \geq \langle m | y - x \rangle + f(x). \quad (2.11)$$

- llamamos “**subdiferencial**” de f en el punto x al conjunto (vacío o no vacío) de todos los vectores subgradientes de f en x , el cuál será denotado por $\partial f(x)$

Diremos que f es subdiferenciable en x si $\partial f(x) \neq \emptyset$.

La figura 2.2 presenta una interpretación geométrica del subgradiente de una función f . El teorema que viene a continuación establece la conexión entre los operadores monótonos maximales y las funciones en $\Gamma_0(\mathbb{R}^n)$.

Teorema 2.3 (Moreau). Sea $f \in \Gamma_0(\mathbb{R}^n)$. Entonces ∂f es un operador monótono maximal.

Demostración. Ver [Heinz H. Bauschke, 2017]. □

Existen algunas observaciones interesantes con respecto a este último teorema que vale la pena mencionar:

- (1) Puede suceder que $\partial f(\hat{x}) = \emptyset$ y por lo tanto $\hat{x} \notin \text{dom}(\partial f)$ aunque $\hat{x} \in \text{dom}(f)$.
- (2) Cuando $\partial f(\hat{x})$ es un singletón, diremos que f es *diferenciable* en \hat{x} .
- (3) La inversa del teorema (2.3) no es verdad, es decir; no todo operador monótono en \mathbb{R}^n es subgradiente de alguna función en $\Gamma_0(\mathbb{R}^n)$.

La primera observación es la que es relevante para nuestro estudio, pues dada una función $f : \mathbb{R}^n \rightarrow \mathbb{R}$, nos gustaría garantizar que $\text{dom}(\partial f) \neq \emptyset$. Afortunadamente en la clase $\Gamma_0(\mathbb{R}^n)$ de las funciones sobre las cuáles estamos interesados, este problema es subsanado por el teorema que viene a continuación.

Teorema 2.4. Sea $f \in \Gamma_0(\mathbb{R}^n)$. Entonces $\text{dom}(\partial f) \neq \emptyset$

Demostración. Ver [Heinz H. Bauschke, 2017]. □

A continuación presentamos el resultado más importante de este capítulo, este muestra que los minimizadores globales de las funciones en $\Gamma_0(\mathbb{R}^n)$ pueden ser caracterizados por un simple pero potente principio, el cuál; tiene su origen en el siglo XVII en una obra de Pierre Fermat.

Teorema 2.5 (Fermat's rule). Sea $f \in \Gamma_0(\mathbb{R}^n)$. Entonces

$$\text{argmin}(f) = \text{zer}(\partial f) = \{x \in \mathbb{R}^n \mid 0 \in \partial f(x)\}.$$

Demostración. Sea $x \in \mathbb{R}^n$, entonces $x \in \text{argmin}(f) \Leftrightarrow (\forall y \in \mathbb{R}^n) \langle y - x | 0 \rangle + f(x) \leq f(y) \Leftrightarrow 0 \in \partial f(x)$. □

La regla de Fermat nos demuestra que existe una conexión entre la teoría de los operadores monótonos y la optimización convexa. Por lo tanto, gracias a este teorema el problema de minimizar $f \in \Gamma_0(\mathbb{R}^n)$ se puede reducir a la búsqueda de los “zeros” del operador monótono maximal ∂f . Más aún, los algoritmos para la búsqueda de los “zeros” de ∂f , y en general, de cualquier operador monótono maximal $\mathcal{T} : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ pueden ser derivados a partir del algoritmo del punto fijo. La siguiente sección presenta el algoritmo del punto fijo y hace un repaso de algunos algoritmos más importantes que se desprenden a partir de este.

2.4. La iteración del Punto Fijo

Recordemos que \tilde{x} es un punto fijo de un operador $T : \mathbb{R}^n \rightarrow \mathbb{R}^n$ si

$$\tilde{x} = T(\tilde{x}).$$

Más formalmente tenemos la siguiente definición.

Definición 2.14. El conjunto de los puntos fijos de un operador $T : \mathbb{R}^n \rightarrow \mathbb{R}^n$ es denotado por $\text{fix}(T)$, i.e.,

$$\text{fix}(T) = \{x \in \mathbb{R}^n \mid T(x) = x\}.$$

De aquí tenemos que la iteración del algoritmo del punto fijo toma la forma:

Algoritmo del Punto Fijo

1. Iniciar:
 - Elegir $x_0 \in \mathbb{R}^n$
 - $k = 0$.
2. Hasta la convergencia:

Hacer:

 - $x_{k+1} = T(x_k)$
 - $k = k + 1$
 - Controlar el criterio de paro:

Fin de hacer

El algoritmo anterior, llamado también la “*iteración de Picard*”, se remonta a 1890 (ver [Picard, 1890]) y su uso implica dos pasos:

- El primero, es encontrar una función T adecuada cuyos puntos fijos sean soluciones del problema en cuestión que queremos resolver.
- El segundo, es mostrar que la iteración en realidad converge a un punto fijo.

A continuación daremos dos condiciones generales que garantizan la convergencia y a partir de los cuáles se pueden derivar diferentes algoritmos conocidos en la optimización convexa. Sin embargo, debemos advertir que estos no son los únicos enfoques posibles.

2.4.1. Condiciones para la convergencia del algoritmo de Punto Fijo

Condición del Operador Contractivo (contractive operator)

Sea (X, d) un espacio métrico completo y supongamos que $T : X \rightarrow X$ es una contracción, entonces tenemos el siguiente teorema.

Teorema 2.6 (Banach-Picard). *Sea (X, d) un espacio métrico completo (en particular vale cuando $X = \mathbb{R}^n$) y sea $T : X \rightarrow X$ una contracción (i.e. Lipschitz continua con constante $L \in (0, 1)$). Dado un $x_0 \in X$ y si*

$$(\forall k \in \mathbb{N}) \quad x_{k+1} = T(x_k). \quad (2.12)$$

Entonces, existe $\tilde{x} \in X$ tal que los siguientes resultados se cumplen:

- (i) \tilde{x} es el único punto fijo de T .
- (ii) $(\forall k \in \mathbb{N}) \quad d(x_{k+1}, \tilde{x}) \leq Ld(x_k, \tilde{x})$.
- (iii) $(\forall k \in \mathbb{N}) \quad d(x_k, \tilde{x}) \leq L^k d(x_0, \tilde{x})$ (de aquí, x_k converge linealmente a \tilde{x}).
- (iv) $(\forall k \in \mathbb{N}) \quad \frac{d(x_0, x_1)}{1+L} \leq d(x_0, \tilde{x}) \leq \frac{d(x_0, x_1)}{1-L}$.
- (v) Estimación del error a priori: $(\forall k \in \mathbb{N}) \quad d(x_k, \tilde{x}) \leq \frac{L^k}{1-L} d(x_0, x_1)$.
- (vi) Estimación del error a posteriori: $(\forall k \in \mathbb{N}) \quad d(x_k, \tilde{x}) \leq \frac{1}{1-L} d(x_k, x_{k+1})$.

Demostración. Ver [Almezel et al., 2014]. □

Condición del Operador Ponderado (averaged operator)

Supongamos que $T : \mathbb{R}^n \rightarrow \mathbb{R}^n$ es un operador ponderado entonces, la iteración del punto fijo es conocido también como la iteración de *Krasnosel'skii-Mann (KM)* (ver [Mann, 1953]), el cuál garantiza la convergencia del punto fijo siempre que este exista.

Teorema 2.7 (Krasnosel'skii-Mann Iteration). *Sea $\alpha \in (0, 1)$ y supongamos que el operador $T : \mathbb{R}^n \rightarrow \mathbb{R}^n$ es α -averaged tal que $\text{fix}(T) \neq \emptyset$. Sean $\{\lambda_k\}_{k \in \mathbb{N}}$ una sucesión en $[0, 1/\alpha]$ tal que $\sum_{k \in \mathbb{N}} \lambda_k(1 - \alpha\lambda_k) = +\infty$ y $x_0 \in \mathbb{R}^n$. Dada la sucesión $\{x_k\}_{k \in \mathbb{N}}$ generada por la iteración*

$$(\forall k \in \mathbb{N}) \quad x_{k+1} = x_k + \lambda_k(Tx_k - x_k). \quad (2.13)$$

Entonces, se cumplirá lo siguiente:

(i) La sucesión $\{Tx_k - x_k\}_{k \in \mathbb{N}}$ converge a 0.

(ii) La sucesión $\{x_k\}_{k \in \mathbb{N}}$ converge a un punto en $\text{fix}(T)$.

Demostración. Ver [Heinz H. Bauschke, 2017]. □

Debemos notar que cuando $T : \mathbb{R}^n \rightarrow \mathbb{R}^n$ es no-expansivo pero no ponderado, entonces la iteración del punto fijo no necesariamente converge a un punto en $\text{fix}(T)$, incluso cuando $\text{fix}(T) \neq \emptyset$. Un ejemplo simple se puede observar cuando T es una rotación sobre una recta, o una reflexión sobre un plano. En este caso, podemos entonces usar el operador $G = (1 - \theta)I + \theta T$ con $\theta \in (0, 1)$ en la iteración del punto fijo para encontrar los puntos fijos de T .

Corolario 2.1. Sea $T : \mathbb{R}^n \rightarrow \mathbb{R}^n$ un operador firmemente no-expansivo tal que $\text{fix}(T) \neq \emptyset$. Sean $\{\lambda_k\}_{k \in \mathbb{N}}$ una sucesión en $[0, 2]$ tal que $\sum_{k \in \mathbb{N}} \lambda_k(2 - \lambda_k) = +\infty$ y $x_0 \in \mathbb{R}^n$. Dada la sucesión $\{x_k\}_{k \in \mathbb{N}}$ generada por la iteración

$$(\forall k \in \mathbb{N}) \quad x_{k+1} = x_k + \lambda_k(Tx_k - x_k). \quad (2.14)$$

Entonces, se cumplirá lo siguiente:

(i) La sucesión $\{Tx_k - x_k\}_{k \in \mathbb{N}}$ converge a 0.

(ii) La sucesión $\{x_k\}_{k \in \mathbb{N}}$ converge a un punto en $\text{fix}(T)$.

Demostración. Ver [Heinz H. Bauschke, 2017]. □

Resumimos esta sección resaltando que, cuando el operador $T : \mathbb{R}^n \rightarrow \mathbb{R}^n$ es *contractivo*, α -*averaged* o *firmemente no-expansivo*, entonces es posible encontrar el punto fijo asociado a este operador siguiendo las directrices planteadas en 2.6, 2.7 y 2.1 respectivamente. A continuación presentamos algunos algoritmos que se desprenden a partir de estas condiciones:

2.4.2. Método del Gradiente

Consideremos el problema

$$\underset{x \in \mathbb{R}^n}{\text{Minimizar}} \quad f(x),$$

donde $f : \mathbb{R}^n \rightarrow \mathbb{R}$ es una función CCP¹. Asumamos que f es diferenciable, por lo tanto x será una solución si y solo si

$$\nabla f(x) = 0 \quad \Leftrightarrow \quad x = (I - \alpha \nabla f)(x),$$

para $\alpha \in \mathbb{R} \setminus \{0\}$. Vemos rápidamente que x es una solución si y solo si este es un punto fijo del operador $I - \alpha \nabla f$, por lo tanto, el algoritmo de punto fijo para este caso será:

¹Designa a las funciones continuas, cerradas y propias.

Algoritmo: Método del Gradiente

1. Iniciar:
 - Elegir $x_0 \in \mathbb{R}_n$
 - $k = 0$.
2. Hasta la convergencia:
Hacer:
 - $x_{k+1} = x_k - \rho_k \nabla f(x_k)$
 - $k = k + 1$
 - Controlar el criterio de paro:Fin de hacer

Este algoritmo que se remonta al siglo XVIII (ver [Cauchy, 1847]), es conocido también como el método de Cauchy, descenso del gradiente o descenso más pronunciado; aquí ρ_k viene a ser el tamaño del paso para cada iteración.

Podemos dar algunas condiciones que debe cumplir f para la convergencia de dicho algoritmo. Empecemos con el caso cuando $\rho_k = \alpha (\forall k \in \mathbb{N})$, tenemos la siguiente proposición:

Proposición 2.2. *Sea f fuertemente convexa y “strongly smooth”² con parámetros m y L respectivamente, entonces:*

- $I - \alpha \nabla f$ es Lipschitz con parámetro $L_{GM} = \max\{|1 - \alpha m|, 1 - \alpha L|\}$.
- $I - \alpha \nabla f$ es una contracción para $\alpha \in (0, 2/L)$.

Demostración. Ver [Ryu & Boyd, 2016]. □

Esta proposición garantiza que $I - \alpha \nabla f$ es una contracción para ciertas condiciones de f y por lo tanto gracias al teorema 2.6 tenemos que existe una solución \hat{x} y que además el método del gradiente converge linealmente con tasa de convergencia

$$\|x_k - \hat{x}\| \leq L_{GM}^k \|x_0 - \hat{x}\|.$$

En general, la proposición anterior se puede extender para el caso cuando la sucesión de pasos ya no es una constante. En este caso, el método convergerá para cada sucesión de pasos $\{\rho_k\} \subseteq [\varepsilon, 2L - \varepsilon]$, donde $0 < \varepsilon \leq L$ es fijado previamente.

2.4.3. El algoritmo Forward

El primer algoritmo para la búsqueda de los zeros del operador T es el conocido *algoritmo de Forward*, esta es simplemente una generalización de la versión del método del gradiente aplicado a T . Consideremos entonces el problema de encontrar un $x \in \mathbb{R}^n$ que satisfaga

²Designamos con este nombre a una función CCP f que es diferenciable y cuya ∇f es Lipschitz con parametro L .

$$T(x) = 0,$$

donde $T : \mathbb{R}^n \rightarrow \mathbb{R}^n$. Por el mismo argumento anterior, x es una solución si y solo si este es un punto fijo de $I - \alpha T$ para $\alpha \in \mathbb{R} \setminus \{0\}$, por lo tanto, el algoritmo de punto fijo para este caso será:

Algoritmo Forward

1. Iniciar:

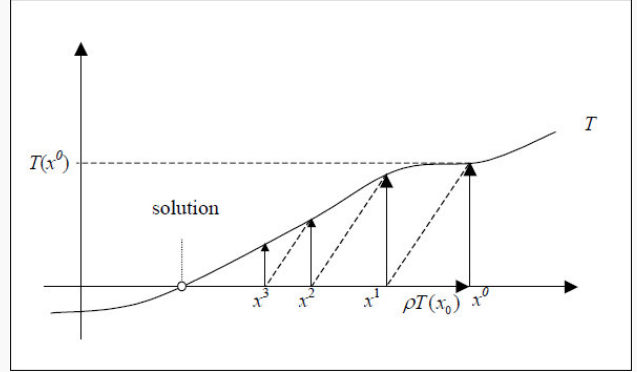
- Elegir $x_0 \in \mathbb{R}^n$
- $k = 0$.

2. Hasta la convergencia:
Hacer:

- $x_{k+1} = x_k - \rho_k T(x_k)$
- $k = k + 1$
- Controlar el criterio de paro:

Fin de hacer

Interpretación geométrica



Aquí también $\{\rho_k\}_{k=0}^{\infty} \subseteq \langle 0, \infty \rangle$ es una sucesión de “stepsizes”. Análogamente al caso anterior podemos dar algunas condiciones para la convergencia.

Proposición 2.3. *Supongamos que T es un operador fuertemente monótono y Lipschitz con parámetros m y L respectivamente, y entonces:*

- *El operador $I - \alpha T$ es lipschitziano de parámetro $L_{GM} = 1 - 2\alpha m + \alpha^2 L^2$.*
- *El operador $I - \alpha T$ es una contracción para $\alpha \in (0, 2m/L^2)$.*

Demostración. Ver [Ryu & Boyd, 2016]. □

Sin embargo, este resultado es más débil que el que teníamos para el método de gradiente, pues los valores de α para los que converge la iteración son más restrictivos, el factor de contracción es peor para todos los valores de $\alpha > 0$. Además, este método en general no convergerá sin el supuesto de la monotonicidad fuerte.

Desafortunadamente, la convergencia de todos estos algoritmos exige condiciones demasiado restrictivas, estos necesitan en particular que el operador en cuestión sea una aplicación, excluyendo de este modo a los operadores subdiferenciales resultantes de la optimización no diferenciable y peor aún, a los operadores multivaluados. No obstante, estos pueden ser fabricados artificialmente por el procedimiento clásico de regularización de *Moreau-Yosida*. Veamos a continuación de que trata todo esto.

2.5. Construcción del algoritmo del punto proximal generalizado para ∂f

Retomemos el problema de minimizar $f \in \Gamma_0(\mathbb{R}^n)$, es decir:

$$\underset{x \in \mathbb{R}^n}{\text{Minimizar}} \quad f(x),$$

donde en general f es no diferenciable. Como ya hemos comentado, este problema se reduce a la búsqueda de los “zeros” del operador monótono maximal ∂f , es decir, encontrar

$$\hat{x} \in \mathbb{R}^n / 0 \in \partial f(\hat{x}).$$

Como f es no diferenciable, entonces el operador monótono maximal ∂f es multivaluado. Para poder aplicar los algoritmos estudiados en la sección anterior, debemos entonces asociar a este operador su resolvente y su aproximación de Yosida. Estos se definen a partir del concepto de *convolución*.

Definición 2.15 (*Infimal Convolution*). Sean f y g funciones definidas de \mathbb{R}^n en $(-\infty, +\infty]$. La **infimal convolution** (llamado también “epi-sum”) de f y g es:

$$\begin{aligned} f \square g : \mathbb{R}^n &\rightarrow \overline{\mathbb{R}} \\ z &\mapsto \inf_{x \in \mathbb{R}^n} [f(x) + g(z - x)], \end{aligned} \tag{2.15}$$

y decimos que este es **exacta en el punto** z si:

$$(f \square g)(z) = \min_{x \in \mathbb{R}^n} [f(x) + g(z - x)],$$

en consecuencia, existe $x \in \mathbb{R}^n$ tal que

$$(f \square g)(z) = f(x) + g(z - x) \in (-\infty, +\infty]. \tag{2.16}$$

Diremos que $f \square g$ es simplemente **exacta** si este es axacta en cada punto de su dominio, y lo denotaremos por $f \boxdot g$.

Un ejemplo central de la “infimal convolution” es aquel obtenido usando la potencia de una norma, la proposición que viene a continuación ilustra este caso.

Proposición 2.4. Sean $f : \mathbb{R}^n \rightarrow (-\infty, +\infty]$ una función propia³ y $p \in [1, +\infty)$. Consideremos la convolución infimal

$$(\forall \lambda \in \mathbb{R}_{++}) \quad \Psi_\lambda = f \square \left(\frac{1}{\lambda p} \|\cdot\|^p \right). \tag{2.17}$$

Entonces, se cumplirán las siguientes propiedades para cada $\lambda \in \mathbb{R}_{++}$ y cada $z \in \mathbb{R}^n$:

- (i) $\text{dom}(\Psi_\lambda) = \mathbb{R}^n$.
- (ii) $\inf \Psi_\lambda(\mathbb{R}^n) = \inf f(\mathbb{R}^n)$.

³Decimos que una función $f : \mathbb{R}^n \rightarrow \overline{\mathbb{R}}$ es propia si $-\infty \notin f(\mathbb{R}^n)$ y $\text{dom}(f) \neq \emptyset$.

(iii) $\inf \Psi_\lambda(z) \downarrow \inf f(\mathbb{R}^n)$ cuando $\lambda \uparrow +\infty$.

Demostración. Ver [Heinz H. Bauschke, 2017]. \square

Proposición 2.5. Sean $f \in \Gamma_0(\mathbb{R}^n)$, $\lambda \in \mathbb{R}_{++}$ y $p \in (1, +\infty)$, entonces la convolución infimal Ψ_λ definida en (2.17) es convexa, propia, continua y exacta. Más aún, para cada $z \in \mathbb{R}^n$ el ínfimo que define a Ψ_λ para cada z es único.

Demostración. Ver [Heinz H. Bauschke, 2017]. \square

El caso más importante y el que nos servirá en esta tesis es cuando $p = 2$, el cuál recibe el nombre de *envolvente de Moreau-Yosida*, es denotado por $\mathcal{M}_f^\lambda(z)$ y tiene la forma:

$$\mathcal{M}_f^\lambda(z) = \inf \left(f(x) + \frac{1}{2\lambda} \|x - z\|^2 \right) \quad (2.18)$$

donde $\lambda \in \mathbb{R}_{++}$. Como un caso especial de (2.17), la envolvente de Moreau hereda todas las propiedades recogidas en las proposiciones (2.4) y (2.5), es decir, la función a minimizar en (2.18) es fuertemente convexa, no infinita y por lo tanto posee un único mínimo para cada $z \in \mathbb{R}^n$ (incluso cuando $\text{dom}(f) \subsetneq \mathbb{R}^n$). Más precisamente tenemos el siguiente teorema.

Teorema 2.8 (Moreau-1965). Sea $f \in \Gamma_0(\mathbb{R}^n)$. Entonces para cada $z \in \mathbb{R}^n$ y $\lambda \in \mathbb{R}_{++}$ tenemos que la función numérica

$$x \rightarrow f(x) + \frac{1}{2\lambda} \|x - z\|^2$$

posee un mínimo estricto.

Demostración. Se desprende directamente de la proposición (2.5). \square

A partir de este resultado tenemos la siguiente definición:

Definición 2.16. Sea $f \in \Gamma_0(\mathbb{R}^n)$.

- Definimos el **Operador Proximal** de parámetro λ como:

$$\mathcal{P}_f^\lambda(z) = \arg \min_{x \in \mathbb{R}^n} f(x) + \frac{1}{2\lambda} \|x - z\|^2 \quad (2.19)$$

- Definimos la **envolvente de Moreau-Yosida** de parámetro λ como:

$$\mathcal{M}_f^\lambda(z) = \min_{x \in \mathbb{R}^n} f(x) + \frac{1}{2\lambda} \|x - z\|^2 \quad (2.20)$$

Notemos que el *operador proximal* \mathcal{P}_f^λ devuelve el único punto que alcanza el ínfimo que define a \mathcal{M}_f^λ , en consecuencia,

$$\mathcal{M}_f^\lambda(z) = \min_{x \in \mathbb{R}^n} f(x) + \frac{1}{2\lambda} \|x - z\|^2 = f(\mathcal{P}_f^\lambda(z)) + \frac{1}{2\lambda} \|\mathcal{P}_f^\lambda(z) - z\|^2$$

para cada $z \in \mathbb{R}^n$.

2.5.1. Interpretación geométrica del operador Proximal

La figura 2.3 muestra lo que hace un operador proximal. Las líneas negras finas son curvas de nivel de una función convexa f ; la línea negra más gruesa indica el límite de su dominio. La evaluación de \mathcal{P}_f^λ en los puntos azules, los mueve hacia los puntos rojos correspondientes. Los tres puntos en el dominio de la función, permanecen en el dominio y se mueven hacia el mínimo de la función, mientras que los otros dos se mueven hacia el límite del dominio y hacia el mínimo de la función. El parámetro λ controla el grado con la que el operador proximal mapea los puntos hacia el mínimo de f . Cuánto más grande es λ , los puntos mapeados se acercarán más al mínimo, y valores más pequeños darán un movimiento más pequeño hacia el mínimo. La definición indica que $\mathcal{P}_f^\lambda(z)$ es un punto que minimiza f manteniéndose cerca de z , por esta razón, \mathcal{P}_f^λ es a veces llamado el *punto proximal* de z con respecto a f . El parámetro λ puede interpretarse como un peso relativo o *parametro de compensación* entre ambos términos.

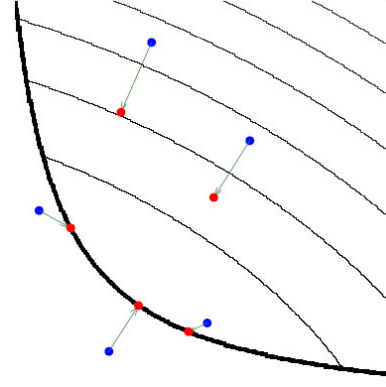


Figura 2.3: Evaluando un operador proximal en varios puntos.

Cuando f es la función indicador

$$1_C(x) = \begin{cases} 0, & x \in C \\ +\infty, & x \notin C \end{cases}$$

donde C es un subconjunto convexo, cerrado y no vacío de \mathbb{R}^n ; entonces el operador proximal de f se reduce a la **proyección** sobre C , el cuál denotamos

$$\Pi_C(z) = \arg \min_{x \in C} \|x - z\|.$$

2.5.2. Interpretación geométrica de la envolvente de Moreau-Yosida

La envolvente de Moreau-Yosida es esencialmente una forma regularizada de f , este tiene como $\text{dom}(\mathcal{M}_f^\lambda) = \mathbb{R}^n$, incluso cuando f no lo tiene, y es continuamente diferenciable, incluso cuando f no lo es. Además, los conjuntos de minimizadores de f y \mathcal{M}_f^λ son los mismos. Los problemas de minimizar f y \mathcal{M}_f^λ son, por lo tanto, equivalentes y este último es siempre un problema de optimización diferenciable con la advertencia de que \mathcal{M}_f^λ puede ser difícil de evaluar. De hecho, algunos algoritmos para minimizar f se interpretan mejor como algoritmos para minimizar \mathcal{M}_f^λ como veremos más adelante. La gráfica 2.4 muestra gráficamente esta interpretación.

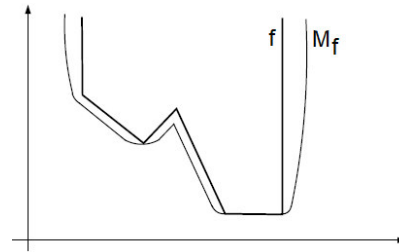


Figura 2.4: Interpretación gráfica de la envolvente de Moreau-Yosida.

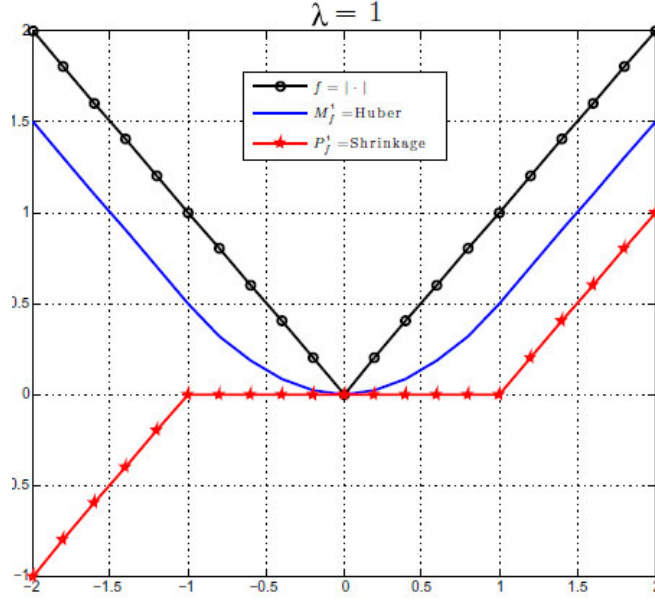


Figura 2.5: La envolvente de Moreau y el operador proximal de la función valor absoluto.

A manera de ejemplo, tomemos $\mathbb{R}^n = \mathbb{R}$ y $f(x) = |x|$. A partir de esta función obtenemos el operador shrinkage y la función de Huber (ver la figura 2.5):

$$\mathcal{P}_f^\lambda(z) = \left(1 - \frac{\lambda}{\|z\|}\right)_+ z$$

$$\mathcal{M}_f^\lambda(z) = \begin{cases} \frac{1}{2\lambda} \|z\|^2, & \|z\| \leq \lambda \\ \|z\| - \frac{\lambda}{2}, & \|z\| \geq \lambda \end{cases}$$

Proposición 2.6. Sea $f \in \Gamma_0(\mathbb{R}^n)$, entonces los operadores \mathcal{P}_f^λ y $\mathcal{I} - \mathcal{P}_f^\lambda$ son firmemente no-expansivos.

Demostración. Ver [Heinz H. Bauschke, 2017]. □

La proposición (2.6) junto con el corolario (2.1) garantizan la existencia de los puntos fijos de los operadores \mathcal{P}_f^λ y $\mathcal{I} - \mathcal{P}_f^\lambda$, más aún, tenemos los siguientes teoremas:

Teorema 2.9. Sea $f \in \Gamma_0(\mathbb{R}^n)$. Para cada $\lambda \in \mathbb{R}_{++}$ tenemos:

$$\text{fix}(\mathcal{P}_f^\lambda) = \text{argmin}(f) \tag{2.21}$$

Demostración. Ver [Heinz H. Bauschke, 2017]. □

Teorema 2.10 (Rockafellar). Sea $f \in \Gamma_0(\mathbb{R}^n)$ y $\lambda \in \mathbb{R}_{++}$. Entonces, la función $\mathcal{M}_f^\lambda : \mathbb{R}^n \rightarrow \mathbb{R}$ es diferenciable en \mathbb{R}^n , y su gradiente

$$\nabla \mathcal{M}_f^\lambda = \lambda^{-1}(\mathcal{I} - \mathcal{P}_f^\lambda) \tag{2.22}$$

es Lipschitz continua de parámetro λ^{-1} .

Demostración. Ver [Heinz H. Bauschke, 2017]. □

Un análisis más detallado muestra que el operador \mathcal{P}_f^λ en realidad está muy relacionado con el operador monótono maximal ∂f del siguiente modo:

$$\begin{aligned} z = (\mathcal{I} + \lambda \partial f)^{-1} &\Leftrightarrow x \in z + \lambda \partial f(z) \\ &\Leftrightarrow 0 \in \partial_z (\lambda f(z) + \tfrac{1}{2} \|z - x\|^2) \\ &\Leftrightarrow z = \arg \min_u (f(u) + \tfrac{1}{2\lambda} \|u - x\|^2) \end{aligned}$$

donde f es convexa y $\lambda \in \mathbb{R}_{++}$ (Otra forma de ver que $\arg \min$ es único es notar que la función al lado derecho es estrictamente convexa). En general, tenemos la siguiente proposición.

Proposición 2.7. Sea $f \in \Gamma_0(\mathbb{R}^n)$, y sean $x, p \in \mathbb{R}^n$. Entonces

$$p = \mathcal{P}_f^\lambda(x) \Leftrightarrow x - p \in \lambda \partial f(p).$$

En otras palabras,

$$\mathcal{P}_f^\lambda = (\mathcal{I} + \lambda \partial f)^{-1}. \quad (2.23)$$

El operador $\mathcal{R} = (\mathcal{I} + \lambda \partial f)^{-1}$ es llamado la **resolvente** de ∂f y en este caso coincide con \mathcal{P}_f^λ . La idea del operador resolvente es la que nos permitirá generalizar estos conceptos a la teoría de los operadores monótonos maximales, la versión general lo revisaremos brevemente en la próxima sección. La aplicación del algoritmo (2.4.2) al operador \mathcal{M}_f^λ cuando $\lambda = 1$ es conocido a veces como el algoritmo del subgradiente y tiene la forma:

Algoritmo: Método del Subgradiente

1. Iniciar:
 - Elegir $x_0 \in \mathbb{R}_n$
 - $k = 0$.
2. Hasta la convergencia:

Hacer:

 - $x_{k+1} = x_k - \rho_k \mathcal{M}_f(x_k) = (1 - \rho_k)x_k + \rho_k \mathcal{P}_f(x_k)$
 - $k = k + 1$
 - Controlar el criterio de paro:

Fin de hacer

2.6. Resolvente de Operadores Monótonos

Antes de introducir los métodos generales recordemos que los “zeros” de un operador monótono $\mathcal{T} : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ es denotado por:

$$\text{zer}(\mathcal{T}) = \mathcal{T}^{-1}(0).$$

Proposición 2.8. *Sea $\mathcal{T} : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ un operador estrictamente monótono o fuertemente monótono de parámetro m . Entonces $\text{zer}(\mathcal{T})$ es a lo más un singleton.*

Demostración. Ver [Heinz H. Bauschke, 2017]. □

Consideremos el operador multivaluado $\mathcal{T} : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$. Tenemos la siguiente definición:

Definición 2.17. *Sea $\alpha \in \mathbb{R}$.*

- La **resolvente** de \mathcal{T} esta dada por

$$\mathcal{R}_{\mathcal{T}} = (\mathcal{I} + \alpha \mathcal{T})^{-1}. \quad (2.24)$$

- El **operador Caley** de \mathcal{T} esta dada por

$$\mathcal{C}_{\mathcal{T}} = 2\mathcal{R}_{\mathcal{T}} - \mathcal{I}. \quad (2.25)$$

Proposición 2.9. *Sea $\alpha > 0$, tenemos lo siguiente:*

- (i) *Si \mathcal{T} es monótono entonces $\mathcal{R}_{\mathcal{T}}$ y $\mathcal{C}_{\mathcal{T}}$ son funciones no-expansivas.*
- (ii) *Si \mathcal{T} es monótono maximal entonces $\text{dom}(\mathcal{R}_{\mathcal{T}}) = \text{dom}(\mathcal{C}_{\mathcal{T}}) = \mathbb{R}^n$.*
- (iii) *$0 \in \mathcal{T}$ si y solo si $x = \mathcal{R}_{\mathcal{T}}(x) = \mathcal{C}_{\mathcal{T}}(x)$.*

Demostración. Ver [Ryu & Boyd, 2016]. □

Dos preguntas que surgen inmediatamente son las siguientes: ¿Cuándo $\mathcal{R}_{\mathcal{T}}$ será una contracción?, análogamente ¿Cuándo $\mathcal{C}_{\mathcal{T}}$ será una contracción?. La proposición que viene a continuación responde a estas preguntas.

Proposición 2.10.

- *Si \mathcal{T} es fuertemente monótono de parámetro m , entonces $\mathcal{R}_{\mathcal{T}}$ es una contracción con constante de Lipschitz $L_R = 1/(1 + \alpha m)$.*
- *Si \mathcal{T} es fuertemente monótono y Lipschitz de parámetros m y L respectivamente, entonces $\mathcal{C}_{\mathcal{T}}$ es una contracción con constante de Lipschitz $L_C = (1 - \frac{4\alpha m}{(1+\alpha L)^2})^{1/2}$.*

Demostración. Ver [Ryu & Boyd, 2016]. □

Debemos tener cuidado que si \mathcal{T} es fuertemente monótono sin ninguna otra condición, entonces $\mathcal{C}_{\mathcal{T}}$ no necesariamente será una contracción.

Método del Punto Proximal

Consideremos el problema de encontrar un x que satisfaga

$$0 \in \mathcal{T}(x),$$

donde \mathcal{T} es un operador monótono maximal. Tenemos los siguientes algoritmos:

- **Algoritmo de Cayley:** Por el ítem(iii) de la proposición 2.9 sabemos que, $0 \in \mathcal{T}(x)$ si y solo si $x = C_{\mathcal{T}}(x)$ cuando $\alpha > 0$ y si además $C_{\mathcal{T}}$ es ponderado entonces por el teorema 2.7 la iteración

$$(\forall k \in \mathbb{N}) \quad x_{k+1} = C_{\mathcal{T}}(x_k)$$

convergerá hacia un punto fijo si es que este existe (notemos que la condición de no-expansividad garantizada por el ítem(i) de la proposición 2.9 no es suficiente para garantizar la convergencia).

- **Algoritmo del Punto Proximal:** Por el ítem(iii) de la proposición 2.9 sabemos que, $0 \in \mathcal{T}(x)$ si y solo si $x = \mathcal{R}_{\mathcal{T}}(x)$ cuando $\alpha > 0$ entonces por el teorema 2.7 la iteración

$$(\forall k \in \mathbb{N}) \quad x_{k+1} = \mathcal{R}_{\mathcal{T}}(x_k)$$

convergerá hacia un punto fijo si es que este existe (notemos que aquí $\mathcal{R}_{\mathcal{T}}$ es un operador ponderado).

La maximalidad del operador juega un rol muy importante en este contexto. En efecto, la iteración del punto fijo para una función arbitraria F puede volverse indefinida en algún paso cuando este escapa del $\text{dom}(F)$, sin embargo, gracias al ítem(ii) de la proposición 2.9 tenemos que $\text{dom}(\mathcal{R}_{\mathcal{T}}) = \text{dom}(C_{\mathcal{T}}) = \mathbb{R}^n$ lo cual exime de este problema a los algoritmos que acabamos de presentar.

Capítulo 3

Operadores Splitting y Algoritmos distribuidos

En este capítulo, nuestro objetivo es encontrar un cero de un operador monótono que admita una división en dos operadores monótonos maximales. En otras palabras, queremos encontrar un $x \in \mathbb{R}^n$ tal que $0 \in (\mathcal{T}_1 + \mathcal{T}_2)(x)$, donde $\mathcal{T}_1, \mathcal{T}_2 : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ son dichos operadores monótonos maximales. La idea obviamente es transformar este problema en una ecuación de punto fijo con operadores contruidos a partir de $\mathcal{T}_1, \mathcal{T}_2$, sus resolventes y sus operadores de Cayley vistos en la sección 2.6 del capítulo anterior. Debemos mencionar que estos métodos son útiles solo cuando los operadores implicados se pueden calcular de manera eficiente.

3.1. El algoritmo de división de Peaceman-Rachford

Consideremos el problema de encontrar un x que satisfice:

$$0 \in (\mathcal{T}_1 + \mathcal{T}_2)(x), \quad (3.1)$$

donde \mathcal{T}_1 y \mathcal{T}_2 son operadores multivaluados monótonos maximales tal que $\text{zer}(\mathcal{T}_1 + \mathcal{T}_2) \neq \emptyset$. Para todo $\alpha > 0$ tenemos que

$$0 \in (\mathcal{T}_1 + \mathcal{T}_2)(x) \Leftrightarrow C_{\mathcal{T}_1} C_{\mathcal{T}_2}(z) = z, \quad x = \mathcal{R}_{\mathcal{T}_2}(z),$$

donde $C_{\mathcal{T}_1}$ y $C_{\mathcal{T}_2}$ son los operadores de Caley de \mathcal{T}_1 y \mathcal{T}_2 respectivamente y además, $\mathcal{R}_{\mathcal{T}_2}$ es la resolvente de \mathcal{T}_2 , los cuáles dependen de α y fueron definidos en 2.17. Para demostrar esta última relación podemos ayudarnos de la siguiente identidad.

Lema 3.1. *Si \mathcal{T} es un operador multivaluado monótono maximal y $\alpha > 0$ entonces se cumplirá:*

$$C_{\mathcal{T}}(I + \alpha\mathcal{T}) = I - \alpha\mathcal{T}$$

Demostración. Notemos que las suposiciones hechas hacen que $(I + \alpha\mathcal{T})^{-1}$ sea una función. De este modo para cada $x \in \text{dom}(\mathcal{T})$ tenemos:

$$\begin{aligned} C_{\mathcal{T}}(I + \alpha\mathcal{T})(x) &= 2(I + \alpha\mathcal{T})^{-1}(I + \alpha\mathcal{T})(x) - (I + \alpha\mathcal{T})(x) \\ &= 2I(x) - (I + \alpha\mathcal{T})(x) \\ &= (I - \alpha\mathcal{T})(x). \end{aligned}$$

Para concluir la prueba, observemos que si $x \notin \text{dom}(\mathcal{T})$, entonces ambos lados de la igualdad serán conjuntos vacíos. \square

Gracias a este lema ahora es fácil demostrar que:

$$\begin{aligned}
0 \in \mathcal{T}_1(x) + \mathcal{T}_2(x) &\Leftrightarrow 0 \in (\mathcal{I} + \alpha\mathcal{T}_1)x - (\mathcal{I} - \alpha\mathcal{T}_2)x \\
&\Leftrightarrow 0 \in (\mathcal{I} + \alpha\mathcal{T}_1)x - \mathcal{C}_{\mathcal{T}_2}(\mathcal{I} + \alpha\mathcal{T}_2)x \\
&\Leftrightarrow 0 \in (\mathcal{I} + \alpha\mathcal{T}_1)x - \mathcal{C}_{\mathcal{T}_2}(z), \quad z \in (\mathcal{I} + \alpha\mathcal{T}_2)x \\
&\Leftrightarrow \mathcal{C}_{\mathcal{T}_2}(z) \in (\mathcal{I} + \alpha\mathcal{T}_1)\mathcal{R}_{\mathcal{T}_2}(z), \quad x = \mathcal{R}_{\mathcal{T}_2}(z) \\
&\Leftrightarrow \mathcal{R}_{\mathcal{T}_1}\mathcal{C}_{\mathcal{T}_2}(z) = \mathcal{R}_{\mathcal{T}_2}(z), \quad x = \mathcal{R}_{\mathcal{T}_2}(z) \\
0 \in (\mathcal{T}_1 + \mathcal{T}_2)(x) &\Leftrightarrow \mathcal{C}_{\mathcal{T}_1}\mathcal{C}_{\mathcal{T}_2}(z) = z, \quad x = \mathcal{R}_{\mathcal{T}_2}(z)
\end{aligned}$$

De aquí, dado $z^0 \in \mathbb{R}^n$, la iteración del punto fijo será:

Peaceman-Rachford iteration

- ▶ $x^{k+1/2} = \mathcal{R}_{\mathcal{T}_2}(z^k)$
- ▶ $z^{k+1/2} = 2x^{k+1/2} - z^k$
- ▶ $x^{k+1} = \mathcal{R}_{\mathcal{T}_1}(z^{k+1/2})$
- ▶ $z^{k+1} = 2x^{k+1} - z^{k+1/2}$

Este método fue presentado por primera vez en [Kellogg, 1969; Lions & Mercier, 1979; Peaceman & Rachford, 1955] y es conocido con el nombre de *división de Peaceman-Rachford*. La convergencia de este algoritmo requiere condiciones adicionales, pues para $\alpha > 0$ solo se puede garantizar la no-expansividad del mapeo $\mathcal{C}_{\mathcal{T}_1}\mathcal{C}_{\mathcal{T}_2}$, de este modo, sin estas suposiciones adicionales la iteración no necesariamente convergerá. Por ejemplo, si \mathcal{T}_2 es uniformemente monótona, entonces se puede demostrar que la sucesión generada por dichas iteraciones converge fuertemente a una solución (ver [Heinz H. Bauschke, 2017]).

3.2. El algoritmo de división de Douglas-Rachford

Consideremos nuevamente el problema propuesto en (3.1). Para garantizar la convergencia y subsanar la debilidad del algoritmo de *división de Peaceman-Rachford*, ponderamos el mapeo no expansivo $\mathcal{C}_{\mathcal{T}_1}\mathcal{C}_{\mathcal{T}_2}$ con la identidad \mathcal{I} . Claramente, para todo $\alpha > 0$ tendremos:

$$0 \in (\mathcal{T}_1 + \mathcal{T}_2)(x) \Leftrightarrow \left(\frac{1}{2}\mathcal{I} + \frac{1}{2}\mathcal{C}_{\mathcal{T}_1}\mathcal{C}_{\mathcal{T}_2} \right)(z) = z, \quad x = \mathcal{R}_{\mathcal{T}_2}(z).$$

Por lo tanto, dado $z^0 \in \mathbb{R}^n$ la iteración del punto fijo tiene la forma:

Douglas-Rachford iteration

- $x^{k+1/2} = \mathcal{R}_{\mathcal{T}_2}(z^k)$
- $z^{k+1/2} = 2x^{k+1/2} - z^k$
- $x^{k+1} = \mathcal{R}_{\mathcal{T}_1}(z^{k+1/2})$
- $z^{k+1} = z^k + x^{k+1} - x^{k+1/2}$

Este método fue presentado en [Douglas & Rachford, 1956; Lions & Mercier, 1979] y es conocido con el nombre de *división de Douglas-Rachford*. Del capítulo anterior sabemos que tanto $\mathcal{C}_{\mathcal{T}_1}$ como $\mathcal{C}_{\mathcal{T}_2}$ son siempre no-expansivas, por lo tanto la convergencia geométrica esta garantizada cuando al menos uno de los operadores de Caley es una contracción. En [Heinz H. Bauschke, 2017] se da una prueba rigurosa de la convergencia de este algoritmo, en el cuál también se analiza su extensión para una suma finita de operadores multivaluados monótonos maximales.

3.3. El algoritmo de división Forward-Backward

Los dos primeros algoritmos presentados valen para operadores multivaluados monótonos maximales en general. Focalizemonos ahora en el caso cuando $\mathcal{T}_1 = T$ es univariado, es decir un operador monótono maximal en el sentido habitual.

Consideremos el problema de encontrar un x que satisfice:

$$0 \in (T + \mathcal{T})(x). \quad (3.2)$$

donde T y \mathcal{T} son operadores monótonos maximales. Observemos la diferencia en el tipo de letra, la primera representa a un operador en el sentido habitual y la segunda a un operador multivaluado. Para todo $\alpha > 0$ tenemos que

$$\begin{aligned} 0 \in (T + \mathcal{T})(x) &\Leftrightarrow 0 \in (I + \alpha\mathcal{T})(x) - (I - \alpha T)(x) \\ &\Leftrightarrow (I - \alpha T)(x) \in (I + \alpha\mathcal{T})(x) \\ &\Leftrightarrow x = (I + \alpha\mathcal{T})^{-1}(I - \alpha T)(x) \end{aligned}$$

De este modo, x será una solución si y solo si este es un punto fijo de $\mathcal{R}_{\mathcal{T}}(I - \alpha T)(x)$. Dado $x^0 \in \mathbb{R}^n$, la iteración del punto fijo tiene la forma reducida:

Forward-Backward iteration

- $x^{k+1} = \mathcal{R}_{\mathcal{T}}(x^k - \alpha T(x^k)).$

Este método fue presentado por primera vez en [Passty, 1979] y es conocido con el nombre de algoritmo de *división de Forward-Backward*. Existen condiciones interesantes para la convergencia de este algoritmo. Si por ejemplo asumimos que T es el operador subdiferencial con constante de Lipschitz L y si $\alpha \in (0, 2/L)$. O también, si asumimos que T es fuertemente monótona y Lipschitz con parámetros m y L respectivamente y si $\alpha \in (0, 2m/L^2)$. Entonces, del capítulo anterior tendremos que el paso *forward* $(I - \alpha T)$ será *ponderado*. El paso *backward* $(I + \alpha T)^{-1}$ es ponderado para todo $\alpha > 0$. Así, cuando $(I - \alpha T)$ es *ponderado*, la composición $(I + \alpha T)^{-1}(I - \alpha T)$ es un operador *ponderado* y por lo tanto la sucesión generada por las iteraciones de dicho algoritmo convergerá a una solución si este existe. Una condición más general para la convergencia cuando trabajamos en espacios de Hilbert es que T sea un operador *cocoercivo*. Este algoritmo es el análogo del algoritmo forward y backward de Euler válido para resolver ecuaciones diferenciales.

3.3.1. El algoritmo del Gradiente Proximal

Un caso particular del algoritmo de *división de Forward-Backward* se presenta cuando deseamos resolver el problema

$$\underset{x \in \mathbb{R}^n}{\text{Minimizar}} \quad f(x) + g(x) \quad (3.3)$$

donde $f, g \in \Gamma_0(\mathbb{R}^n)$ con la condición adicional que f es diferenciable. Debemos advertir que esta forma de *división* no es única, por lo que diferentes tipos de *división* conducirán a diferentes implementaciones del método de gradiente proximal para el mismo problema original.

Para poder deducir un algoritmo para este problema, notemos que \hat{x} es una solución de (3.3), i.e., este minimiza $f + g$ si y solo si

$$0 \in \nabla f(\hat{x}) + \partial g(\hat{x}).$$

Por lo tanto, para cada $\alpha > 0$, esta condición de optimalidad se cumple si y solo si se cumplen las siguientes equivalencias:

$$0 \in \alpha \nabla f(\hat{x}) + \alpha \partial g(\hat{x})$$

$$0 \in \alpha \nabla f(\hat{x}) - \hat{x} + \hat{x} + \alpha \partial g(\hat{x})$$

$$(I - \alpha \nabla f)(\hat{x}) \in (I + \alpha \partial g)(\hat{x})$$

$$\hat{x} = (I + \alpha \partial g)^{-1}(I - \alpha \nabla f)(\hat{x})$$

$$\hat{x} = \text{prox}_{\alpha g}(\hat{x} - \alpha \nabla f(\hat{x})).$$

Donde si comparamos con el algoritmo Forward-Backward, tendríamos que $\mathcal{R}_{\partial g} = \text{prox}_{\alpha g}$ y $T = \nabla f$. De aquí, tenemos que \hat{x} minimiza $f + g$ si y solo si este es un punto fijo del operador *forward-backward*

$$(I + \alpha \partial g)^{-1}(I - \alpha \nabla f).$$

Este método, llamado *método del gradiente proximal* aplica repetidamente este operador para obtener un punto fijo y, por lo tanto, una solución al problema original. La condición $\alpha \in (0, 1/L]$, donde L es una constante de Lipschitz de ∇f garantiza que el operador *forward-backward* es

ponderado y, por lo tanto, la iteración converge a un punto fijo (cuando uno existe). Podemos dar explícitamente el algoritmo, el cuál toma la forma:

**Algoritmo:
Gradiente Proximal**

1. Iniciar:
 - Elegir $x^0 \in \mathbb{R}^n$, $\beta \in (0, 1)$, α^0
 - $\alpha := \alpha^{k-1}$.
2. Hasta la convergencia:
Hacer:
 - $z := \text{prox}_{\alpha g}(x^k - \alpha \nabla f(x^k))$
 - break si $f(z) \leq \tilde{f}_\alpha(z, x^k)$.
 - Actualizar $\alpha := \beta \alpha$
 Fin de hacer
3. $\alpha^k := \alpha$, $x^{k+1} := z$.

Existen dos elementos nuevos que aparecen en este algoritmo, el parámetro β y la función \tilde{f} . Respecto al parámetro, un valor bastante aceptable con el cuál trabaja muy bien el algoritmo es con $\beta = 1/2$, respecto a la función \tilde{f} , esta es definida como:

$$\tilde{f}_\alpha(x, y) = f(x) + \nabla f(y)^t(x - y) + \frac{1}{2\alpha} \|x - y\|_2^2,$$

con $\alpha > 0$. Para y fijo, esta función es convexa y satisface $\tilde{f}_\alpha(x, x) = f(x)$, y es una cota superior de f cuando $\alpha \in (0, 1/L]$, donde L es la constante de Lipschitz para ∇f (ver [Combettes & Pesquet, 2011; Karimi *et al.*, 2016]).

Existe una versión simple para la aceleración del algoritmo del gradiente proximal, esta se obtiene del siguiente modo:

$$\begin{aligned} y^{k+1} &= x^k + \omega^k(x^k - x^{k-1}) \\ x^{k+1} &= \text{prox}_{\alpha^k g}(y^{k+1} - \alpha^k \nabla f(y^{k+1})), \end{aligned}$$

donde $\omega^k \in [0, 1)$ es un parámetro de extrapolación y α^k es el *step size*. Los parámetros ω^k deben elegirse de manera específica para lograr la aceleración de la convergencia. Una opción simple sugerida por L. Vandenberghe (click aquí) es tomar

$$\omega^k = \frac{k}{k+3}.$$

Respecto a los pasos α^k , cuando ∇f es Lipschitz continua con constante L , se puede mostrar que este método converge con una tasa $O(1/k^2)$ cuando se usa un tamaño de paso fijo $\alpha^k = \alpha \in (0, 1/L]$. Si L es desconocido, los tamaños de los pasos α^k se eligen en cada iteración. A continuación

mostramos el algoritmo acelerado con una elección de los pasos propuesta en [Beck & Teboulle, 2009], el cuál tiene la forma

**Algoritmo:
Gradiente Proximal Acelerado**

1. Iniciar:
 - Elegir x^0 , $\beta \in (0, 1)$, α^0
 - $\alpha := \alpha^{k-1}$.
2. Hasta la convergencia:

Hacer:

 - $y^k = x^k + \frac{k}{k+3}(x^k - x^{k-1})$;
 - $z := \text{prox}_{\alpha g}(y^k - \alpha \nabla f(y^k))$
 - break si $f(z) \leq \tilde{f}_\alpha(z, y^k)$.
 - Actualizar $\alpha := \beta\alpha$

Fin de hacer
3. $\alpha^k := \alpha$, $x^{k+1} := z$.

Aquí, la función \tilde{f} es la misma definida para el algoritmo anterior. Más detalles y otras versiones para la aceleración de este algoritmo pueden ser encontradas en [Nesterov *et al.*, 2007; Tseng, 2008].

3.4. El algoritmo ADMM (Método de dirección alternada de multiplicadores)

En esta sección desarrollaremos en profundidad el algoritmo ADMM, un algoritmo simple pero potente que se adapta bien a la optimización convexa distribuida, y en particular a los problemas que surgen en las estadísticas aplicadas y la inteligencia artificial. Este también toma la forma de un procedimiento de descomposición-coordinación y puede verse como un intento de combinar los beneficios de la descomposición dual y los métodos del *Lagrangiano aumentado* para la optimización restringida.

Vale la pena enfatizar que el algoritmo en sí no es nuevo, pues fue introducido por primera vez a mediados de la década de 1970 en [Gabay & Mercier, 1975; Glowinski & Marroco, 1975], aunque ideas bastante similares surgieron desde mediados de la década de 1950. El algoritmo se estudió a lo largo de la década de 1980 y, a mediados de la década de 1990, se habían establecido casi todos los resultados teóricos que mencionaremos aquí. El hecho de que el ADMM se haya desarrollado mucho antes de la disponibilidad de los sistemas informáticos distribuidos a gran escala y los problemas de optimización masiva explicaría por qué no es tan conocido, sin embargo, cada vez va ganando más popularidad hoy en día.

3.4.1. Deducción del algoritmo

Consideremos el problema

$$\begin{aligned} & \text{Minimizar} && f(x) + g(z) \\ & \text{s.a.} && \\ & && Ax + Bz = c, \end{aligned} \tag{3.4}$$

donde $x \in \mathbb{R}^n$ y $z \in \mathbb{R}^m$ son las variables de optimización, $A \in \mathbb{R}^{p \times n}$, $B \in \mathbb{R}^{p \times m}$ y $c \in \mathbb{R}^p$; además, $f \in \Gamma_0(\mathbb{R}^n)$ y $g \in \Gamma_0(\mathbb{R}^m)$. Denotemos el valor óptimo del problema (3.4) por

$$p^* = \inf\{f(x) + g(z) \mid Ax + Bz = c\}.$$

El problema dual asociado será entonces

$$\text{Maximizar} \quad -f^*(-A^t y) - g^*(-B^t y) + c^t y, \tag{3.5}$$

donde $y \in \mathbb{R}^p$ y f^* y g^* representan las conjugadas de f y g .

El operador sufderivacional de la función dual

$$F(y) = -A \partial f^*(-A^t y) - B \partial g^*(B^t y) - c$$

admite la división $F = F_1 + F_2$, donde

$$F_1(y) = -A \partial f^*(-A^t y) - c, \quad y \quad F_2(y) = -B \partial g^*(B^t y).$$

Asumamos que f y g son estrictamente convexos, entonces tenemos que los arg min están bien definidos. Aplicando *Douglas-Rachford* a $F = F_1 + F_2$ tenemos:

$$\begin{aligned} \zeta^{k+1} &= \mathcal{R}_{F_2}(v^k) \\ \xi^{k+1} &= \mathcal{R}_{F_1}(2\zeta^{k+1} - v^k) \\ v^{k+1} &= v^k + \xi^{k+1} - \zeta^{k+1}. \end{aligned}$$

Vimos en el capítulo anterior que la evaluación de las resolventes implica un paso de minimización. Explicitando esto obtenemos:

$$\begin{aligned} \tilde{z}^{k+1} &= \arg \min_z \left(g(z) + (v^k)^t Bz + \frac{\alpha}{2} \|Bz\|_2^2 \right) \\ \zeta^{k+1} &= v^k + \alpha B \tilde{z}^{k+1} \\ \tilde{x}^{k+1} &= \arg \min_x \left(f(x) + (v^k + 2\alpha B \tilde{z}^{k+1})^t (Ax - c) + \frac{\alpha}{2} \|Ax - c\|_2^2 \right) \\ \xi^{k+1} &= v^k + \alpha (A \tilde{x}^{k+1} - c) + 2\alpha B \tilde{z}^{k+1} \\ v^{k+1} &= v^k + \alpha (A \tilde{x}^{k+1} + B \tilde{z}^{k+1} - c). \end{aligned}$$

Como las iteraciones de ζ^k y ξ^k ya no tienen ninguna dependencia explícita, estos se pueden eliminar. Substituyendo $v^k = \alpha u^k + \alpha (A \tilde{x}^k - c)$ tenemos

$$\tilde{z}^{k+1} = \arg \min_z \left(g(z) + \frac{\alpha}{2} \|A\tilde{x}^k + Bz - c + u^k\|_2^2 \right)$$

$$\tilde{x}^{k+1} = \arg \min_x \left(f(x) + \frac{\alpha}{2} \|Ax + B\tilde{z}^{k+1} - c + u^{k+1}\|_2^2 \right)$$

$$u^{k+1} = u^k + A\tilde{x}^k + B\tilde{z}^{k+1} - c.$$

Finalmente, cambiamos el orden de u^{k+1} y \tilde{x}^{k+1} para obtener la dependencia correcta y sustituir $\tilde{x}^k = x^{k+1}$ y $\tilde{z}^k = z^k$, y así obtener el algoritmo ADMM, que toma finalmente la forma:

ADMM iteration 1^{ra} forma

- $x^{k+1} = \arg \min_x \left(f(x) + \frac{\alpha}{2} \|Ax + Bz^{k+1} - c + u^k\|_2^2 \right)$
- $z^{k+1} = \arg \min_z \left(g(z) + \frac{\alpha}{2} \|Ax^{k+1} + Bz - c + u^k\|_2^2 \right)$
- $u^{k+1} = u^k + Ax^{k+1} + Bz^{k+1} - c.$

Debemos señalar que hay otras formas de analizar el ADMM, por ejemplo, en [Yan & Yin, 2016] se puede obtener el ADMM aplicando el algoritmo de *división de Douglas-Rachford* al problema de optimización primal, por otro lado, un enfoque que evita de algún modo la teoría de los operadores monótonos podemos encontrarlo en [Boyd *et al.*, 2011; Gabay & Mercier, 1975; Glowinski & Marroco, 1975]. Aquí, hemos seguido el enfoque de [Gabay, 1983], aplicando la *división de Douglas-Rachford* al problema dual. No obstante, podemos explicitar la relación existente con el método de los multiplicadores de Lagrange, para esto, formemos el *Lagrangiano aumentado*

$$\mathcal{L}_\rho(x, z, y) = f(x) + g(z) + y'(Ax + Bz - c) + \frac{\rho}{2} \|Ax + Bz - c\|_2^2, \quad \text{con } \rho > 0,$$

en este caso, el ADMM tendrá la forma

ADMM iteration 2^{da} forma

- $x^{k+1} = \arg \min_x \mathcal{L}_\rho(x, z^k, y^k)$
- $z^{k+1} = \arg \min_z \mathcal{L}_\rho(x^{k+1}, z, y^k)$
- $y^{k+1} = u^k + \rho(Ax^{k+1} + Bz^{k+1} - c),$

donde $\rho > 0$. En esta segunda versión se hacen presentes tanto la variable primal como la dual, más aún, aquí (z^{k+1}, y^{k+1}) es una función de (z^k, y^k) . La variable x^k es un resultado intermedio que

viene calculado del paso previo a (z^{k-1}, y^{k-1}) .

La relación existente entre la primera y la segunda versión no es difícil de observar, para esto es suficiente definir el residuo como $r = Ax + Bz - c$, de donde obtenemos

$$\begin{aligned} y^t r + (\rho) \|r\|_2^2 &= \frac{\rho}{2} \|r + \frac{1}{\rho} y\|_2^2 - \frac{1}{2\rho} \|y\|_2^2 \\ &= \frac{\rho}{2} \|r + u\|_2^2 - \frac{\rho}{2} \|u\|_2^2. \end{aligned}$$

donde $u = \frac{1}{\rho} y$ es la *variable dual escalada*. Reemplazando estos valores obtenemos la primera versión del algoritmo el cuál también es conocido como la *forma escalada* del algoritmo ADMM.

3.4.2. Análisis de la convergencia

Existen muchos resultados para la convergencia del algoritmo ADMM, en nuestro caso, nuestra primera versión heredará las condiciones de convergencia del método de Douglas-Rachford ya que fué deducida a partir de este. Sin embargo, podemos hacer una suposición adicional sobre el problema a optimizar (3.4).

Suposición: El *Lagrangiano no aumentado* \mathcal{L}_0 tiene un punto silla.

Explícitamente, esto quiere decir que existe un $(\hat{x}, \hat{z}, \hat{y})$ no necesariamente único, tal que $\forall x, z, y$ se cumple

$$\mathcal{L}_0(\hat{x}, \hat{z}, y) \leq \mathcal{L}_0(\hat{x}, \hat{z}, \hat{y}) \leq \mathcal{L}_0(x, z, \hat{y}).$$

Esta suposición, garantiza que los valores optimales del problema primal y dual coinciden, i.e., la dualidad fuerte se cumple. Por otro lado, notemos que no existen suposiciones adicionales respecto a A , B o c , en particular, no es necesario que las matrices sean *full rank*.

Bajo la suposición hecha y dado que $f \in \Gamma_0(\mathbb{R}^m)$ y $g \in \Gamma_0(\mathbb{R}^n)$, las iteraciones del ADMM satisfacen lo siguiente:

- *Convergencia residual:* $r^k \rightarrow 0$ cuando $k \rightarrow \infty$, i.e., las iteraciones convergen a la factibilidad.
- *Convergencia al objetivo:* $f(x^k) + g(x^k) \rightarrow p^*$ cuando $k \rightarrow \infty$, i.e., las iteraciones de la función objetivo se aproxima al valor óptimo.
- *Convergencia de la variable dual:* $y^k \rightarrow \hat{y}$ cuando $k \rightarrow \infty$, donde \hat{y} es el valor dual óptimo.

Una prueba rigurosa sobre estos resultados podemos encontrarlo por ejemplo en [Boyd *et al.*, 2011]. En la práctica sin embargo, existen ejemplos que muestran que el ADMM puede ser muy lento cuando deseamos que la convergencia sea de alta precisión. No obstante, muy a menudo ocurre que el ADMM converge a una precisión modesta dentro de unas pocas decenas de iteraciones, suficiente para muchas aplicaciones. Este comportamiento hace que el ADMM sea bastante similar a ciertos algoritmos como por ejemplo, el método del gradiente conjugado, el cuál necesita a menudo de unas pocas decenas de iteraciones para producir resultados aceptables de uso práctico.

La convergencia lenta del ADMM también lo distingue de algoritmos como el método de Newton (o, para problemas restringidos, métodos de punto interior), donde se puede lograr una alta precisión en un tiempo razonable. Aunque en algunos casos es posible combinar ADMM con un método para producir una solución de alta precisión a partir de una solución de baja precisión [Wilkinson & GOLUB, 1966], en el caso general, el ADMM será prácticamente útil sobre todo en los casos en que la precisión moderada sea suficiente. Afortunadamente, este suele ser el caso de los tipos de problemas que consideramos, pues resolver un problema de estimación de parámetros con una precisión muy alta a menudo produce poca o ninguna mejora en el rendimiento de la predicción real.

3.4.3. Condiciones para la optimalidad

Las condiciones necesarias y suficientes para la optimalidad del algoritmo ADMM que resuelve (3.4) son:

- la factibilidad primal:

$$A\hat{x} + B\hat{z} - c = 0, \quad (3.6)$$

- la factibilidad dual:

$$0 \in \partial f(\hat{x}) + A^t \hat{y} \quad (3.7)$$

$$0 \in \partial g(\hat{z}) + B^t \hat{y}. \quad (3.8)$$

Cuando f y g son diferenciables, estas subdiferenciales pueden ser reemplazadas por los gradientes ∇f y ∇g , y la notación \in debe ser reemplazada por $=$.

Como z^{k+1} minimiza $\mathcal{L}_\rho(x^{k+1}, z, y^k)$ por definición, tenemos que:

$$\begin{aligned} 0 &\in \partial g(z^{k+1}) + B^t y^k + \rho B^t (Ax^{k+1} + Bz^{k+1} - c) \\ &= \partial g(z^{k+1}) + B^t y^k + \rho B^t r^{k+1} \\ &= \partial g(z^{k+1}) + B^t y^{k+1}. \end{aligned}$$

Esto significa que z^{k+1} y y^{k+1} siempre satisfacen (3.8), por lo que alcanzar la optimalidad se reduce a satisfacer (3.6) y (3.7). Este fenómeno es análogo a las iteraciones del método de los multiplicadores que siempre es factible dual. Análogamente, como x^{k+1} minimiza $\mathcal{L}_\rho(x, z^k, y^k)$ por definición, tenemos que:

$$\begin{aligned} 0 &\in \partial f(x^{k+1}) + A^t y^k + \rho A^t (Ax^{k+1} + Bz^k - c) \\ &= \partial f(x^{k+1}) + A^t (y^k + \rho r^{k+1} + \rho B(z^k - z^{k+1})) \\ &= \partial f(x^{k+1}) + A^t y^{k+1} + \rho A^t B(z^k - z^{k+1}), \end{aligned}$$

o equivalentemente,

$$\rho A^t B(z^{k+1} - z^k) \in \partial f(x^{k+1}) + A^t y^{k+1}.$$

Esto significa que la cantidad

$$s^{k+1} = \rho A^t B(z^{k+1} - z^k)$$

puede ser visto como un residuo para la condición de factibilidad (3.7). Nos referiremos a s^{k+1} como el *residuo dual* respecto a la iteración $k + 1$, y a $r^{k+1} = Ax^{k+1} + Bz^{k+1} - c$ como el *residuo primal* respecto a la iteración $k + 1$.

En resumen, las condiciones de optimalidad para el problema ADMM consiste de las tres condiciones (3.6), (3.7) y (3.8). Por lo tanto, los dos residuos convergerán a cero a medida que el ADMM avanza. De hecho, la prueba de convergencia propuesta en [Boyd *et al.*, 2011] muestra que $B(z^{k+1} - z^k)$ converge a cero, lo que implica que s^k también converge a cero.

3.4.4. Criterio de finalización

Los residuos de las condiciones de optimalidad pueden relacionarse con un límite para la suboptimalidad de la función objetivo en un punto dado del siguiente modo

$$f(x^k) + g(z^k) - p^* \leq -(y^k)^t r^k + (x^k - \hat{x})^t s^k. \quad (3.9)$$

Gracias a esta desigualdad, tenemos que la suboptimalidad de la función objetivo debe ser pequeño cuando los residuos r^k y s^k son pequeños. Sin embargo, no podemos usar directamente esta desigualdad como un criterio de paro, ya que no conocemos \hat{x} . Pero, si de algún modo estimamos que $\|x^k - \hat{x}\|_2 \leq d$, para algún d , tendremos que

$$f(x^k) + g(z^k) - p^* \leq -(y^k)^t r^k + d\|s^k\|_2 \leq \|y^k\|_2 \|r^k\|_2 + d\|s^k\|_2.$$

El término medio y el de la derecha de la desigualdad, pueden ser usados como un límite aproximado de la suboptimalidad para la función objetivo (que depende del valor d aún no hallado). Esto sugiere que un criterio para detener el proceso de iteración razonable es que los residuos primal y dual deben ser pequeños, es decir,

$$\|r^k\|_2 \leq \epsilon^{pri} \quad \text{y} \quad \|s^k\|_2 \leq \epsilon^{dual},$$

donde $\epsilon^{pri} > 0$ y $\epsilon^{dual} > 0$ son tolerancias aceptables para las condiciones de factibilidad primal y dual propuestas en (3.6) y (3.7), respectivamente. Estas tolerancias pueden ser elegidas usando el criterio del valor absoluto y relativo, de modo que

$$\begin{aligned} \epsilon^{pri} &= \sqrt{p}\epsilon^{abs} + \epsilon^{rel} \max\{\|Ax_2^k\|_2, \|Bz^k\|_2, \|c\|_2\}, \\ \epsilon^{dual} &= \sqrt{n}\epsilon^{abs} + \epsilon^{rel}\|A^t y^k\|_2, \end{aligned}$$

donde $\epsilon^{abs} > 0$ es una tolerancia absoluta y $\epsilon^{rel} > 0$ es una tolerancia relativa. De aquí, un valor razonable para el criterio de paro relativo podría ser $\epsilon^{rel} = 10^{-3}$ o 10^{-4} , dependiendo de la aplicación. Por otro lado, la elección del criterio de paro absoluto dependerá de la escala de los valores de las variables típicas. Concluimos esta sección mencionando que existen diversos criterios para la aceleración del algoritmo ADMM dependiendo del problema en específico que se pretende resolver, para nuestros objetivos, este panorama de algún modo general, es suficiente para ilustrar las aplicaciones que presentaremos a continuación.

3.5. Optimización distribuida de consenso con ADMM

A continuación describimos un método simple para obtener algoritmos proximales paralelos y distribuidos para resolver problemas de optimización convexa. El método se basa en el algoritmo ADMM y la clave es dividir la función objetivo (y las restricciones) en dos términos, donde al menos uno de ellos debe ser separable. La separabilidad de los términos nos da la capacidad de poder evaluar el operador proximal en forma paralela.

La separabilidad de una función puede expresarse en forma de una partición, es decir, sea $[n] = \{1, \dots, n\}$, decimos que la colección $\mathcal{P} = \{\varsigma_1, \dots, \varsigma_N\}$ es una **partición** de $[n]$ si $\cup \mathcal{P} = [n]$ y $\varsigma_i \cap \varsigma_j = \emptyset$ para $i \neq j$, donde $\varsigma_i \subseteq [n]$, $\forall i = 1, \dots, N$ y un subvector $x_{\varsigma_i} \in \mathbb{R}^{|\varsigma_i|}$ es referenciado por sus índices en ς_i . Gracias a esta partición tenemos la siguiente definición.

Definición 3.1. Se dice que una función $f : \mathbb{R}^n \rightarrow \mathbb{R}$ es \mathcal{P} -separable si

$$f(x) = \sum_{i=1}^N f_i(x_{\varsigma_i}), \quad (3.10)$$

donde $f_i : \mathbb{R}^{|\varsigma_i|} \rightarrow \mathbb{R}$ y x_{ς_i} es el subvector de x con índices en ς_i .

La definición 3.1 nos dice que f es una suma de los términos f_i donde cada uno de estos depende solo de una parte de x ; si cada $\varsigma_i = \{i\}$ entonces f es completamente separable. Llamaremos a ς_i el alcance de f_i (*scope* en inglés).

La importancia de la separabilidad de una función radica en el hecho de que si f es \mathcal{P} -separable, entonces $(\text{prox}_f(v))_i = \text{prox}_{f_i}(v_i)$, donde $v_i \in \mathbb{R}^{|\varsigma_i|}$, i.e., el operador proximal puede dividirse en N operaciones reducidas que pueden realizarse independientemente en forma paralela, por ejemplo, si f es separable mediante dos variables de modo que $f(x, y) = \varphi(x) + \psi(y)$ entonces tendríamos

$$\text{prox}_f(v, w) = (\text{prox}_\varphi(v), \text{prox}_\psi(w)).$$

Para poder deducir un algoritmo ADMM consideremos el problema

$$\text{Minimizar } f(x) + g(x), \quad (3.11)$$

donde $x \in \mathbb{R}^n$ y $f, g : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ son convexas, cerradas y propias. Asumamos que f y g son \mathcal{P} -separable y \mathcal{Q} -separable respectivamente, donde $\mathcal{P} = \{\varsigma_1, \dots, \varsigma_N\}$ y $\mathcal{Q} = \{d_1, \dots, d_M\}$ son particiones de $[n]$. Escribiendo el problema explícitamente en función de los subvectores tenemos

$$\text{Minimizar } \sum_{i=1}^N f_i(x_{\varsigma_i}) + \sum_{j=1}^M g_j(x_{d_j}), \quad (3.12)$$

donde $f_i : \mathbb{R}^{|\varsigma_i|} \rightarrow \mathbb{R} \cup \{+\infty\}$ y $g_j : \mathbb{R}^{|d_j|} \rightarrow \mathbb{R} \cup \{+\infty\}$, entonces, el algoritmo ADMM para el problema (3.12) tiene la forma

ADMM iteration paralela

- $x_{c_i}^{k+1} := \text{prox}_{\alpha f_i}(z_{c_i}^k - u_{c_i}^k)$
- $z_{d_j}^{k+1} := \text{prox}_{\alpha g_j}(x_{d_j}^{k+1} + u_{d_j}^k)$
- $u^{k+1} := u^k + x^{k+1} - z^{k+1}.$

El primer paso implica N actualizaciones realizadas de forma independiente y en paralelo, cada una de las cuales implica evaluar al operador proximal de uno de los componentes f_i de f , y el segundo paso implica M actualizaciones realizadas de forma independiente en paralelo, cada una involucrando al operador proximal de un componente g_j de g . El paso final, por supuesto, siempre es trivialmente paralelizable. Esto puede visualizarse en la Figura 3.1, el cuál muestra dos particiones de un conjunto de variables. Aquí, la x -update se divide en 3 partes y la z -update se divide en 2 partes. Si por ejemplo, $\mathcal{P} = \mathcal{Q}$, entonces el problema original tiene una función objetivo separable y es trivialmente paralelizable. Por otro lado, las superposiciones en las dos particiones, como se muestra la Figura 3.1, conducirán a la comunicación entre diferentes subsistemas. Por ejemplo, si g no es separable, entonces la z -update implicará la agregación de información a través de las N componentes que pueden manejarse independientemente en la x -update. Esto será más claro a medida que examinemos los casos especiales.

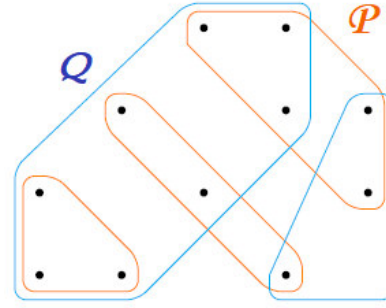


Figura 3.1: Las variables son puntos negros. Las particiones \mathcal{P} y \mathcal{Q} están en naranja y cyan respectivamente.

3.5.1. Protocolo de consenso con ADMM

Consideremos el problema de una función aditiva, i.e., una suma de términos los cuáles comparten una variable en común:

$$\text{Minimizar } f(x) = \sum_{i=1}^N f_i(x),$$

con variable $x \in \mathbb{R}^n$. El problema es minimizar cada uno de los objetivos “locales” f_i , los cuales depende de la misma variable global x . Nuestro objetivo es resolver este problema en modo tal que cada f_i pueda ser manejado en paralelo por un subsistema de procesamiento separado. Para ello, transformamos el problema en una *forma de consenso*:

$$\begin{aligned} &\text{Minimizar} && \sum_{i=1}^N f_i(x_i) \\ &s.a. && x_1 = x_2 = \dots = x_N, \end{aligned} \tag{3.13}$$

con variables $x_i \in \mathbb{R}^n, i = 1, \dots, N$. Es decir, creamos N copias de la variable global original x de tal forma que la función objetivo sea ahora separable, pero agregamos una *restricción de consenso o consistencia* que requiere que todas estas variables “locales” x_i coincidan. Esto puede visualizarse en la Figura 3.2, que muestra un ejemplo con $n = 4$ y $N = 5$; aquí, cada variable local x_i es una columna y las restricciones de consistencia se dibujan a través de filas.

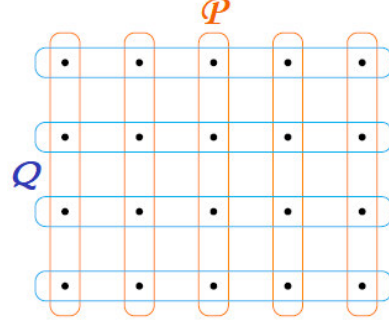


Figura 3.2: Las variables son puntos negros. Las particiones \mathcal{P} y \mathcal{Q} están en naranja y cian respectivamente.

El siguiente paso es transformar (3.13) en la forma (3.11), es decir:

$$\text{Minimizar} \quad \sum_{i=1}^N f_i(x_i) + \iota_C(x_1, \dots, x_N), \quad (3.14)$$

donde C es el *conjunto consenso*

$$C = \{(x_1, \dots, x_N) \mid x_1 = \dots = x_N\}. \quad (3.15)$$

En esta formulación, hemos movido la restricción de consenso hacia el objetivo utilizando la función indicador ι_C ¹. En la notación de (3.11), f es la suma de los términos f_i , mientras que g es la función indicador de la restricción de consistencia. Las particiones son dadas por

$$\mathcal{P} = \{[n], n + [n], 2n + [n], \dots, (N-1)n + [n]\},$$

$$\mathcal{Q} = \{\{i, n + i, 2n + i, \dots, (N-1)n + i\} \mid i = 1, \dots, N\}.$$

La primera partición es clara ya que f es aditiva. La restricción de consenso se divide entre sus componentes, este puede escribirse como una restricción de consenso separada para cada componente. Como la variable completa de optimización para (3.14) está en \mathbb{R}^N , es más fácil verla como en la Figura 3.2, en cuyo caso es fácil ver que f es separable entre columnas mientras que g es separable entre filas.

Ahora aplicamos el algoritmo ADMM. La evaluación de $\text{prox}_{\alpha g}$ se reduce a la proyección sobre el conjunto de consenso (3.15). Esto es simple: reemplazamos cada z_i con su promedio $\bar{z} = \frac{1}{N} \sum_{i=1}^N z_i$.

A partir de esto concluimos que $\sum_{i=1}^N u_i^k = 0$, lo que permite algunas simplificaciones del algoritmo

¹La función indicador de un conjunto convexo C es definido como:

$$\iota_C(u) := \begin{cases} 0 & , \text{ si } u \in C \\ +\infty & , \text{ si } u \notin C \end{cases}$$

general, tomando finalmente la forma

Algoritmo de Consenso Proximal

- $x_i^{k+1} := \text{prox}_{\alpha f_i}(\bar{x}^k - u_i^k)$
- $u_i^{k+1} := u_i^k + x_i^{k+1} - \bar{x}^{k+1}.$

En este *algoritmo de consenso proximal*, cada uno de los N subsistemas realiza de forma independiente una actualización dual y evalúa su operador proximal local; entre éstas, todas las variables locales x_i^k se promedian y el resultado se le pasa a cada subsistema. El método es muy intuitivo: las variables duales u_i (escaladas), que miden la desviación de x_i del promedio \bar{x} , se actualizan de forma independiente para conducir las variables en consenso, y la regularización cuadrática ayuda a que las variables alcancen su valor promedio al mismo tiempo que intenta minimizar cada f_i local.

3.6. Experimentación numérica

Toda la experimentación numérica lo realizaremos en una *pc core i7-7500 de 2.90 GHz de 64 bits con 2 núcleos y 16 GB de RAM*. Mientras que las soluciones referenciales a los problemas planteados serán calculados con el software “CVX”(descargar) desarrollado por *CVX Research, Inc.* Esta es una corporación creada para respaldar el desarrollo de herramientas de optimización convexas incluido el paquete homónimo CVX. Sus desarrolladores, Michael Grant² y Stephen Boyd³ fueron galardonados en el año 2012 con el Premio “*Beale-Hays-Orchard Prize for Excellence in Computational Mathematical Programming*” por su trabajo en CVX.

Con respecto a los algoritmos estudiados en las secciones anteriores, para nuestras experimentaciones numéricas nos focalizaremos principalmente en 3 de ellos: el ADMM, que como ya mencionamos en 3.4 este también puede ser visto como un caso particular del algoritmo de *división de Douglas-Rachford*; el gradiente proximal y su versión acelerada, ambos derivados como casos particulares del algoritmo de *división Forward-Backward*.

Primera experimentación numérica:

Esta primera experimentación, tiene como objetivo mostrar la versatilidad y rapidez de convergencia de los algoritmos de división, para esto tomaremos como modelo de partida la optimización de una forma cuadrática.

Consideremos el problema siguiente:

$$\begin{aligned}
 &\text{Minimizar} && 0.5x^T A x + b^T + c \\
 &\text{s.a.} && L \leq x_i \leq U, \forall i = 1, \dots, n;
 \end{aligned} \tag{3.16}$$

² Ver biografía completa y CV (University of Texas at Austin).

³ Ver biografía completa y CV (Stanford University).

donde $x \in \mathbb{R}^n$, $A \in \mathbb{S}_+^n$, $b \in \mathbb{R}^n$, $c \in \mathbb{R}$ y $L = -50 = -U$. Si queremos implementar los métodos de división sobre este problema, debemos transformarlo a su forma equivalente

$$\begin{aligned} \text{Minimizar} \quad & 0,5x^\tau Ax + b^\tau + c + \iota_{\{z \mid L \leq z \leq U\}} \\ \text{s.a.} \quad & x - z = 0, \end{aligned} \tag{3.17}$$

donde $f(x) = 0,5x^\tau Ax + b^\tau + c$ y $g(z) = \iota_{\{z \mid L \leq z \leq U\}}$ es la función indicador. A partir de esto, vemos que esta estructura tiene la forma estudiada en la sección 3.4. Realizando los cálculos obtenemos:

- Las iteraciones para el ADMM,

$$\begin{aligned} \blacktriangleright \quad x^{k+1} &:= (I + \alpha A)^{-1}(z^k - u^k - \alpha b) \\ \blacktriangleright \quad z^{k+1} &:= \max\{L; \min\{x^{k+1} - u^k; U\}\} \\ \blacktriangleright \quad u^{k+1} &:= u^k + x^{k+1} - z^{k+1}. \end{aligned}$$

- Las iteraciones para el Gradiente proximal,

$$\begin{aligned} \blacktriangleright \quad z^k &:= \max\{L; \min\{x^k - \alpha(0,5(A + A^t)x^k + b); U\}\} \\ \blacktriangleright \quad x^{k+1} &:= z^k. \end{aligned}$$

- Las iteraciones para el Gradiente proximal acelerado,

$$\begin{aligned} \blacktriangleright \quad y^k &:= x^k + \frac{k}{k+3}(x^k - x^{k-1}) \\ \blacktriangleright \quad z^k &:= \max\{L; \min\{y^k - \alpha(0,5(A + A^t)y^k + b); U\}\} \\ \blacktriangleright \quad x^{k+1} &:= z^k. \end{aligned}$$

Consideremos ahora una matriz aleatoria $A \in \mathbb{R}^{500 \times 500}$, para garantizar la definidad positividad podemos realizar el siguiente cálculo $A = A * A^\tau + \sqrt{n}I$, de este modo aseguramos la existencia de un mínimo global, el vector b y el número real c también son aleatorios. La figura 3.3a muestra la velocidad de convergencia hacia la solución óptima de los tres algoritmos. Como se puede observar, el algoritmo ADMM y el gradiente proximal acelerado son los que convergen mas rápidamente a la solución óptima, en este caso el ADMM requirió 53 iteraciones en un tiempo de 0.067582 segundos, mientras que el gradiente proximal acelerado requirió 45 iteraciones con un tiempo de 0.041609 segundos, sin embargo, el algoritmo del gradiente proximal resultó estar más cerca al valor óptimo real con un error del 0.0012 %, mientras que el ADMM tuvo un error del 0.5996 %. La tabla 3.1 reporta estos datos obtenidos.

Después de normalizar las columnas de la matriz A , ejecutamos nuevamente los algoritmos y obtuvimos los datos que se muestran en la tabla 3.2. En este caso el algoritmo ADMM resultó ser abrumadoramente más veloz que los otros, requiriendo solo 4 iteraciones para llegar a la solución óptima con un tiempo de 0.009797 y un error del 0 %. Esto puede ser corroborado en la figura 3.3b. En general, el algoritmo ADMM sacrifica un poco de precisión para alcanzar la solución óptima en

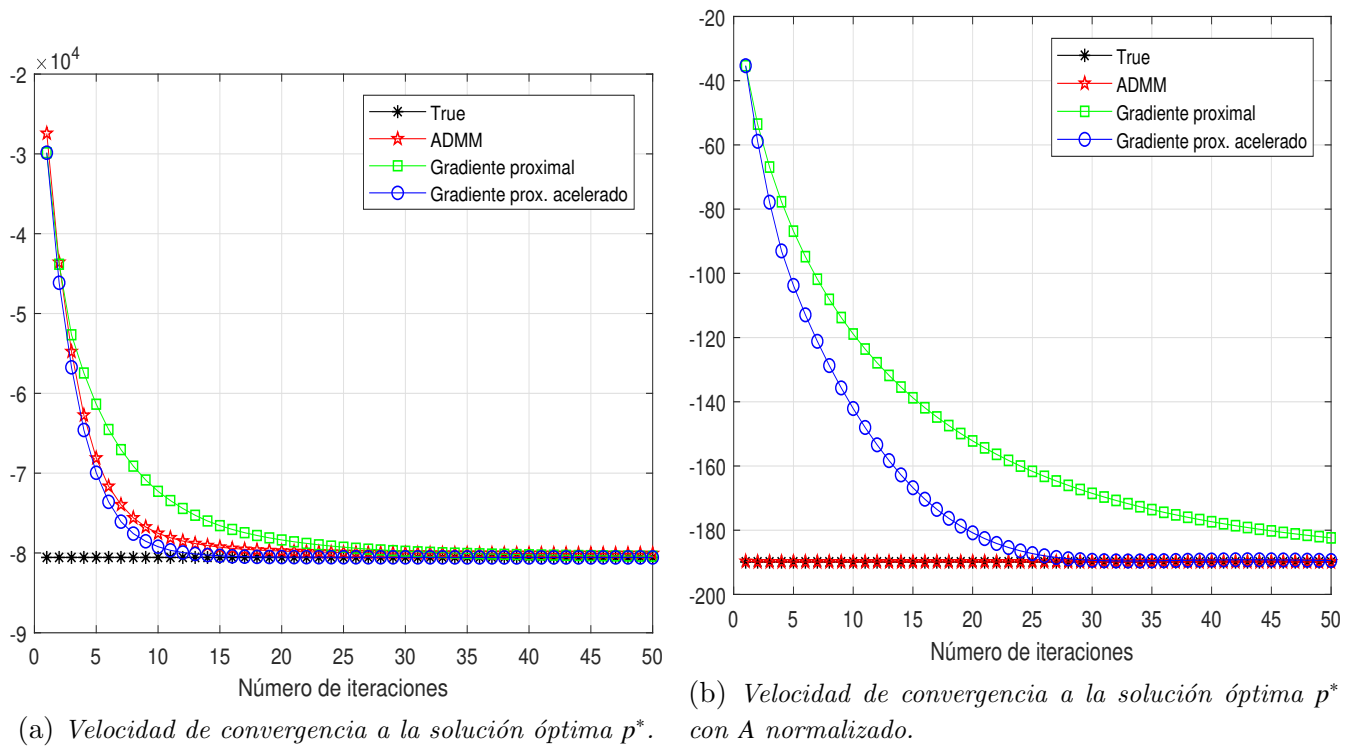


Figura 3.3

Algoritmo	Tiempo(s)	p^*	NumIter
CVX	2.263509	-8.0560e+04	---
ADMM	0.067582	-8.0077e+04	53
G.Prox.	0.099640	-8.0560e+04	211
G.Prox.acel.	0.041609	-8.0559e+04	45

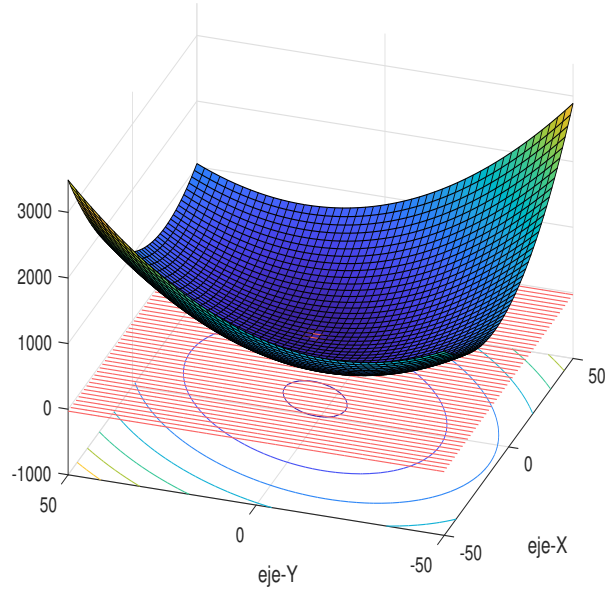
Tabla 3.1: Comparación de los algoritmos para resolver una forma cuadrática con $A \in \mathbb{R}^{500 \times 5000}$.

Algoritmo	Tiempo(s)	p^*	NumIter
CVX	1.695663	-189.8794	---
ADMM	0.009797	-189.8794	4
G.Prox.	0.112196	-189.8371	200
G.Prox.acel.	0.036786	-189.8413	64

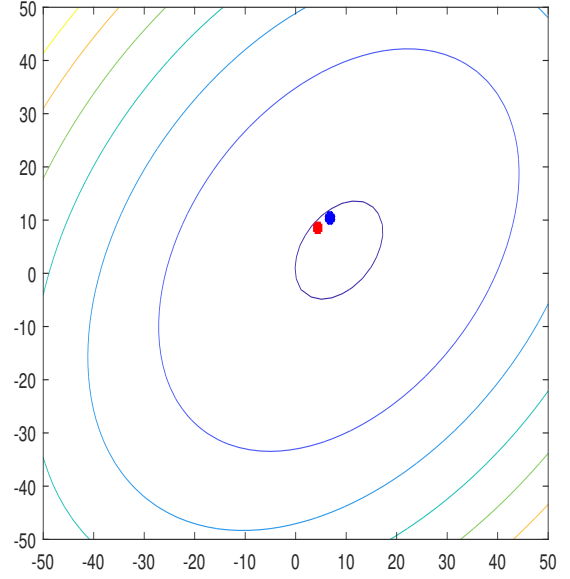
Tabla 3.2: Comparación de los algoritmos para resolver una forma cuadrática con $A \in \mathbb{R}^{500 \times 500}$ con A normalizado.

menor tiempo. Podemos visualizar este fenómeno cuando $A \in \mathbb{R}^{2 \times 2}$, la figura 3.4a muestra la gráfica de la función cuadrática en \mathbb{R}^3 donde se ve claramente que el plano tangente a la solución óptima calculada con el algoritmo ADMM corta ligeramente a la superficie, esto puede ser corroborado en la figura 3.4b, donde el punto azul representa a la solución real y el punto rojo representa a la solución hallada con el ADMM. No obstante, para efectos prácticos esta ligera diferencia no es muy substancial como veremos más adelante.

Concluimos esta experimentación con la tabla 3.3 el cuál muestra los datos obtenidos cuando



(a) Gráfica de la función cuadrática en \mathbb{R}^3 .



(b) Curvas de nivel de la función cuadrática.

Figura 3.4

$A \in \mathbb{R}^{5000 \times 5000}$. Esto corrobora lo que mencionamos en las secciones anteriores, que para valores altos de n , los algoritmos de división, y en particular el ADMM trabaja mejor que los otros algoritmos de la tabla. En la siguiente experimentación profundizaremos un poco más en estos detalles.

<i>Algoritmo</i>	<i>Tiempo(s)</i>	p^*	<i>NumIter</i>
<i>CVX</i>	1052.858279	-411.7942	---
<i>ADMM</i>	1.445512	-411.7937	6
<i>G.Prox.</i>	25.266654	-411.6849	781
<i>G.Prox.acel.</i>	3.605040	-411.6362	102

Tabla 3.3: Comparación de los algoritmos para resolver una forma cuadrática con $A \in \mathbb{R}^{5000 \times 5000}$.

Parte II

Modelización y resolución del problema

Capítulo 4

La gestión de la producción de energía eléctrica sujeto a incertidumbres

Este capítulo tiene como objetivo introducir los complejos problemas inherentes a la gestión de la producción de energía eléctrica, que es el marco para la aplicación industrial de esta tesis. Como se menciona en [Lenoir, 2008], el estudio de los modelos matemáticos para la gestión de la producción eléctrica bajo incertidumbre, pueden ser abordados en el contexto de los problemas de optimización de “grandes sistemas”, es decir, sistemas cuya operación involucra una cantidad enorme de parámetros ([Cohen, 1984], [Lasdon, 2002]). El tamaño a partir del cual un sistema es considerado “grande” es relativamente subjetivo, en donde el consenso consigna esta cualificación a aquellos en el cuál las técnicas de resolución frontal no pueden aplicarse directamente, ya sea porque el tiempo de cálculo es muy grande o simplemente porque el modelo no se puede implantar en la memoria de una computadora. Entonces, es necesario diseñar métodos de resolución alternativos aprovechando las posibles características adicionales del problema. Un ejemplo de este tipo de sistemas es aquel que resulta de la interconexión de múltiples subsistemas enumerados por $1, 2, \dots, p$. Estos subsistemas, son sometidos a decisiones locales y_i :, $i = 1, 2, \dots, p$ en donde el vector $y = (y_1, \dots, y_p)$ representa la variable de decisión global del problema de optimización (ver fig. 4.1).

La decisión y_i a un nivel local genera un costo $F_i(y_i)$ independiente de las decisiones exógenas $y_{j \neq i}$. Si los subsistemas fuesen independientes entre si, el problema de optimización a resolver sería

$$\underset{y}{\text{Minimizar}} \quad F(y) = \sum_{i=1}^p F_i(y_i). \quad (4.1)$$

El inconveniente en este planteamiento surge cuando tal independencia ya no se cumple, es decir, cuando la decisión de cada y_i en realidad genera un efecto en las decisiones óptimas de las otras. Matemáticamente, esta incidencia se puede representar mediante una *función de acoplamiento* $C(\cdot)$ ¹ la cuál dependerá simultáneamente de todas las decisiones y_1, y_2, \dots, y_p , consecuentemente, tendríamos el siguiente problema

$$\underset{y}{\text{Minimizar}} \quad f(y) = \sum_{i=1}^p F_i(y_i) + C(y_1, \dots, y_p) \quad (4.2)$$

¹Esta función recibe el nombre de “fonction couplante” en la terminología francesa.

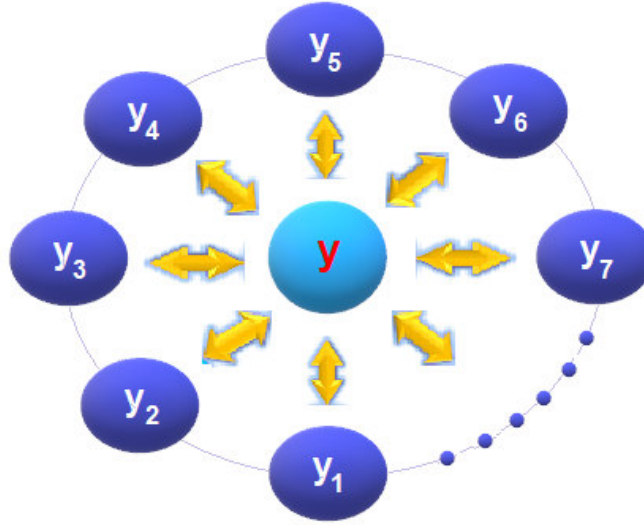


Figura 4.1: Parque de producción eléctrica.

Un método fuertemente usado para abordar problemas de este tipo es conocido con el nombre de “*descomposición-coordinación*”, en donde la idea subyacente es descentralizar la toma de decisiones hacia cada subsistema, en modo que, dichas decisiones sean independientes entre sí como en el caso (4.1). Los detalles de esta técnica lo veremos más adelante, ahora nos centraremos en como podemos usar estos conceptos para contruir un modelo matemático para la gestión de la producción de energía eléctrica.

4.1. Modelización de la gestión de la producción eléctrica

El objetivo aquí consiste en diseñar un modelo matemático que garantice el uso óptimo de un conjunto de plantas de producción eléctrica durante un período determinado. Como mencionamos en la parte introductoria (ver 1.3), el plan de producción que tenemos en mente es a mediano plazo (PPMT). Luego de haber establecido el PPMT, este puede ser ubicado sobre un horizonte temporal de T periodos e indexados por $t = 1, \dots, T$ con duraciones respectivas de $\delta_1, \dots, \delta_T$. Consideremos además que disponemos de una cantidad finita de plantas de producción eléctrica, las cuáles estan indexadas por las letras $u = 1, \dots, U$. Cada una de estas plantas genera durante cada periodo t una potencia $p_{u,t} \in \mathbb{R}$. El costo instantáneo de esta producción podemos entonces representarla por una función $f_{u,t} : \mathbb{R} \rightarrow \mathbb{R}$. Debemos tener en cuenta que las plantas de producción eléctrica son heterogéneas, en el sentido que cada una de ellas funciona de manera diferente de acuerdo a su propia tecnología. En el Perú, contamos con 5 diferentes tipos de tecnología para la producción de dicha energía eléctrica: la termoelectrica, la hidroeléctrica, la eólica, la fotovoltaica y la generación por biomasa. En la siguiente sección se describen de manera breve las características de estas diferentes tecnologías. Por otro lado, como ya mencionamos al inicio de este capítulo, existe un costo C_u el cuál dependerá de la estrategia de producción sobre todo el horizonte temporal $(p_{u,1}, p_{u,2}, \dots, p_{u,T}) \in \mathbb{R}^T$.

Otro aspecto que debemos tomar en cuenta en este punto es la *satisfacción de la demanda*. Es

decir; en cada periodo la potencia total a producir D_t debe ser repartida entre todas las plantas de producción. Las variables de dicha producción estarán por lo tanto ligadas por las restricciones de la demanda, esto podemos escribirlo del siguiente modo:

$$\sum_u p_{u,t} = D_t, \quad \forall t = 1, \dots, T.$$

De aquí, el problema de optimización matemática que acabamos de describir toma la forma:

$$\begin{aligned} & \underset{p_{u,t}}{\text{Minimizar}} \quad \sum_{u,t} f_{u,t}(p_{u,t}) + \sum_u C_u(p_{u,1}, p_{u,2}, \dots, p_{u,T}) \\ & \text{s.a.} \quad \sum_u p_{u,t} = D_t, \quad \forall t = 1, \dots, T. \end{aligned} \quad (4.3)$$

Este problema, puede ser reescrito en una forma más integral apoyandonos de la función indicatriz $C_t : \mathbb{R}^U \rightarrow \mathbb{R}$ definida para cada $t = 1, \dots, T$ del siguiente modo:

$$C_t(p_{1,t}, p_{2,t}, \dots, p_{U,t}) = \begin{cases} 0, & \text{si } \sum_u p_{u,t} = D_t \\ +\infty, & \text{otros casos} \end{cases}$$

el cuál garantiza la satisfacción de la demanda. Luego, el problema (4.3) podemos reescribirlo como:

$$\underset{p_{u,t}}{\text{Minimizar}} \quad \sum_{u,t} f_{u,t}(p_{u,t}) + \sum_u C_u(p_{u,1}, p_{u,2}, \dots, p_{u,T}) + \sum_t C_t(p_{1,t}, p_{2,t}, \dots, p_{U,t}) \quad (4.4)$$

Gracias a esta configuración, la descomposición más intuitiva correspondiente a la noción de sistema físico consiste en considerar las centrales eléctricas como subsistemas en donde la *función de acoplamiento* viene a ser la satisfacción de la demanda, más específicamente:

$$\underset{p_u}{\text{Minimizar}} \quad \sum_u \underbrace{\left[\sum_t f_{u,t}(p_{u,t}) + C_u(p_{u,1}, p_{u,2}, \dots, p_{u,T}) \right]}_{f_u(p_u)} + \underbrace{\sum_t C_t(p_{1,t}, p_{2,t}, \dots, p_{U,t})}_{C(p_1, \dots, p_U)} \quad (4.5)$$

Una alternativa es considerar los periodos en lugar de las centrales como subsistemas, en cuyo caso tendríamos:

$$\underset{p_t}{\text{Minimizar}} \quad \sum_t \underbrace{\left[\sum_u f_{u,t}(p_{u,t}) + C_t(p_{1,t}, p_{2,t}, \dots, p_{U,t}) \right]}_{f_t(p_t)} + \underbrace{\sum_u C_u(p_{u,1}, p_{u,2}, \dots, p_{u,T})}_{C(p_1, \dots, p_T)} \quad (4.6)$$

La tercera descomposición posible es una combinación de las dos precedentes, es decir, una descomposición espacio-temporal que define un subsistema como una central estudiada durante un periodo de tiempo, por lo tanto tendríamos:

$$\underset{p_{u,t}}{\text{Minimizar}} \quad \sum_{u,t} f_{u,t}(p_{u,t}) + \underbrace{\sum_u C_u(p_{u,1}, p_{u,2}, \dots, p_{u,T}) + \sum_t C_t(p_{1,t}, p_{2,t}, \dots, p_{U,t})}_{C(p_{1,1}, \dots, p_{U,T})} \quad (4.7)$$

Como se puede observar, los problemas de optimización (4.5), (4.6) y (4.7) resultan ser un caso particular del problema general (4.2).

4.2. Modelización de las incertidumbres

Los modelos anteriores plantean la cuestión de la adquisición de ciertos datos, como la demanda, los insumos naturales de las centrales hidroeléctricas o los precios de mercado. Por naturaleza, estas cantidades solo se conocen con *incertidumbre* en el momento en el que se plantea el problema de optimización y podrán ser observadas solamente en el futuro. Es importante en este contexto señalar que existen diferentes tipos de incertidumbre; los cuáles son identificados dependiendo de la naturaleza, así como también de la cantidad de información que disponemos. Oberkampff et al. [Oberkampff *et al.*, 2001] propusieron una clasificación de los diferentes tipos de incertidumbre en tres categorías:

- Incertidumbre *aleatorio o estocástico*.
- Incertidumbre *epistémica*.
- *Error*.

La *incertidumbre aleatoria*, es un efecto de una variación inherente del sistema resultado de la naturaleza aleatoria de algunos de sus datos. Si disponemos de un número suficiente de experiencias para deducir explícitamente su ley de probabilidad para dichos datos aleatorios, entonces podremos modelar estos problemas sirviéndonos de los elementos de la teoría de la probabilidad. Por otro lado; la *incertidumbre epistémica*, tiene su origen en la información incompleta del sistema, la ignorancia de ciertos aspectos o la falta de datos experimentales. Esta categoría, puede también ser abordada con las herramientas de la teoría de la probabilidad, sin embargo existe cierta polémica concerniente a la eficacia y la exactitud de este enfoque.

El *error*, de alguna manera puede también ser considerado como un tipo de “*incertidumbre numérico*”. Esta es una falla inevitable, a veces no despreciable que no es el resultado de una falta de información, sino de los límites en la precisión numérica cuando se realizan los cálculos.

La primera categoría, como hemos dicho, puede ser abordada por la teoría de la probabilidad, mientras que la tercera categoría si bien es cierto que no la podemos evitar, al menos podemos estimarla. El verdadero desafío según [Agarwal *et al.*, 2004] es desarrollar una metodología para modelar cuantitativamente la incertidumbre epistémica. El problema más importante con el uso de métodos probabilísticos para tratar problemas donde la incertidumbre es epistémica es que no existen razones particulares para usar una distribución en lugar de otra, siempre que no conozcamos la distribución real que rige el sistema.

En este trabajo, nosotros nos focalizaremos en el estudio de los métodos de optimización que involucren la primera y tercera categoría. Por lo tanto, para modelizar las incertidumbres de nuestro sistema de una forma más general, podemos partir del problema de optimización (4.2), el cuál como vimos involucra como casos particulares a las ecuaciones (4.5), (4.6) y (4.7).

Denotemos entonces por $x_t \in \mathbb{R}^n$ al conjunto de las decisiones tomadas en cada instante t , en consecuencia, estas representarán el conjunto de las producciones de las centrales eléctricas en cada

periodo de tiempo, de aquí, el vector $x = (x_1, \dots, x_T) \in \mathbb{R}^{nT}$ representará la sucesión de decisiones sobre todo el horizonte temporal. El conjunto de los parámetros inciertos observados al inicio de cada período t será representado por la letra $\xi_t \in \mathbb{R}^m$ y por lo tanto el vector $\xi = (\xi_1, \dots, \xi_T) \in \mathbb{R}^{mT}$ representará la sucesión de todas las incertezas a lo largo de todo el horizonte temporal. Denotando a $f : \mathbb{R}^{nT} \times \mathbb{R}^{mT} \rightarrow \mathbb{R}$ como la función objetivo, el problema (4.3) puede ser reescrito de manera abstracta como:

$$\underset{(x_1, \dots, x_T) \in \mathbb{R}^{nT}}{\text{Minimizar}} \quad f(x_1, \dots, x_T; \xi_1, \dots, \xi_T) \quad (4.8)$$

Las incertidumbres sobre los valores de ξ son modelizados matemáticamente con la ayuda de un vector aleatorio definido sobre un espacio de probabilidad $(\Omega, \mathcal{F}, \mathbb{P})$. En realidad, este vector aleatorio viene a ser un proceso estocástico²

$$\begin{aligned} \xi : (\Omega, \mathcal{F}, \mathbb{P}) &\rightarrow (\mathbb{R}^{mT}, \mathcal{B}(\mathbb{R}^{mT})) \\ \omega &\mapsto (\xi_1(\omega), \dots, \xi_T(\omega)) \end{aligned} \quad (4.9)$$

en donde las realizaciones son comunmente llamadas *escenarios* y $\mathcal{B}(\mathbb{R}^{mT})$ es el *sigma*-álgebra de Borel de \mathbb{R}^{mT} . A partir de aquí, la elección de un criterio para resolver este tipo de problemas recae en el campo de la optimización estocástica, por consiguiente, a continuación introduciremos los conceptos generales de dicho tópico.

4.3. Introducción a la Programación Estocástica

Comenzando con los trabajos seminales de Beale, Bellman y Zadeh [Beale, 1955; Bellman, 1957; Bellman & Zadeh, 1970], Charnes, Cooper, Dantzig y Tintner [Charnes *et al.*, 1958; Dantzig, 2010; Tintner, 1955], la optimización bajo incertidumbre ha experimentado un rápido desarrollo en cuanto a la teoría y algoritmos. Dantzig consideraba que la planificación bajo incertidumbre era uno de los problemas abiertos más importantes en la optimización (ver [Horner, 1999]).

En lo sucesivo, hablaremos de incertidumbre en el sentido de *incertidumbre aleatorio* (como ha sido definido por [Oberkampf *et al.*, 2001]), donde una parte de la información necesaria para la comprensión completa de un fenómeno es desconocido. Puede suceder que algunos aspectos del fenómeno se conozcan parcialmente a través de escenarios, datos históricos, una ley de probabilidad, o momentos de la variable aleatoria (por ejemplo, esperanza matemática, varianza). Se pueden crear escenarios a partir de registros históricos (como las ventas recientes, el clima en la última década, la mortalidad de una determinada población), o de la opinión de expertos que pueden predecir el comportamiento del fenómeno incierto. Si la ley de probabilidad de la variable aleatoria es conocida analíticamente, uno puede crear métodos analíticos (la mayoría de las veces esta es una tarea no trivial) que tienen en cuenta esta ley de probabilidad, para generar un conjunto de escenarios según esta ley.

Tomemos entonces un espacio de probabilidad $(\Omega, \mathcal{A}, \mathbb{P})$, donde Ω es el conjunto de las posibles realizaciones ω de los parámetros inciertos, \mathcal{A} un σ -álgebra y \mathbb{P} la correspondiente distribu-

²Ver apéndice B

ción de probabilidad; el problema de optimización estocástica que nos interesa estudiar puede ser representado por:

$$\begin{aligned} & \text{Minimizar} \quad \mathbb{E}[f(x, \omega)] = \int f(x, \omega) d\mathbb{P}[\omega] \\ & \text{s.a} \\ & x \in \mathcal{C} \subseteq \mathbb{R}^n, \end{aligned} \tag{4.10}$$

donde $f(x, \omega)$ es la función objetivo, \mathcal{C} es el conjunto de soluciones factibles definidos por las restricciones del problema de optimización y \mathbb{E} es la esperanza matemática.

A manera de ejemplo, consideremos el caso en el cual Ω es discreto, finito y donde un resultado discreto ω tiene como su correspondiente medida de probabilidad $\mathbb{P}(\omega)$. El problema (4.10) puede ser representado como:

$$\begin{aligned} & \text{Minimizar} \quad \mathbb{E}[f(x, \omega)] = \sum_{\omega \in \Omega} f(x, \omega) \mathbb{P}(\omega) \\ & \text{s.a} \\ & x \in \mathcal{C} \subseteq \mathbb{R}^n. \end{aligned} \tag{4.11}$$

En general, los problemas (4.10) y (4.11) son difíciles de resolver debido al gran número de posibles resultados que debemos tomar en cuenta. Incluso, en el caso más simple como el problema (4.11), existen situaciones prácticas en el cuál el número de elementos en Ω (denotado por $n(\Omega) = |\Omega|$) suele ser muy grande. Para comprender completamente la naturaleza de una solución estocástica, discutiremos a continuación diferentes enfoques para la solución de dicho problema. Por simplicidad conduciremos la discusión para el caso cuando Ω es finito y discreto.

4.3.1. El enfoque “Espera y verás” (Wait and See)

En este enfoque, de alguna manera podemos esperar y ver hasta que la incertidumbre se resuelva al final del horizonte de planificación y se pueda observar un resultado $\omega \in \Omega$ antes de tomar la decisión óptima “ x ”. En consecuencia, el enfoque “*espera y verás*” asume información perfecta sobre el futuro. Es claro que una tal solución no es implementable. Este tipo de problemas pueden expresarse como:

$$\begin{aligned} & z^\omega = \min f(x, \omega) \\ & \text{s.a} \\ & x \in \mathcal{C}^\omega \subseteq \mathbb{R}^n. \end{aligned} \tag{4.12}$$

donde además

$$x^\omega \in \operatorname{argmin}\{f(x, \omega) | x \in \mathcal{C}^\omega\}, \tag{4.13}$$

y

$$z_{\omega s} = \mathbb{E}[z^\omega] = \sum_{\omega} z^\omega \mathbb{P}(\omega). \tag{4.14}$$

El conjunto de las soluciones factibles de x , \mathcal{C}^ω , es definido por las restricciones en el escenario ω . El procedimiento general consiste en resolver el problema para el resultado observado ω y obtener una solución óptima sujeto a dicho escenario, x^ω . Esta es una solución óptima dada la información perfecta sobre el futuro. Luego se calcula el valor esperado del costo óptimo z^ω para obtener $z_{\omega s}$, el cuál viene a ser el costo esperado bajo la información perfecta.

4.3.2. El enfoque “Aquí y ahora”(Here and Now)

Este enfoque representa el verdadero problema de optimización estocástica de 4.11. Una decisión x tiene que ser tomada “*aquí y ahora*”, antes de observar una realización de Ω . El valor x se elige en una manera tal que el costo esperado $\mathbb{E}[f(x, \omega)]$ asuma un valor mínimo, es decir:

$$\begin{aligned} z &= \min \mathbb{E}[f(x, \omega)] \\ \text{s.a} \\ x &\in \mathcal{C} = \cap_{\omega \in \Omega} \mathcal{C}^\omega. \end{aligned} \tag{4.15}$$

El valor óptimo z de la función objetivo denota el costo mínimo esperado del problema de optimización estocástica. Observemos que x tiene que ser feasible para todos los escenarios $\omega \in \Omega$; así $\mathcal{C} = \cap_{\omega \in \Omega} \mathcal{C}^\omega$ denota la intersección de todos los \mathcal{C}^ω , $\omega \in \Omega$, donde \mathcal{C}^ω representa la región feasible dado por las restricciones en el escenario ω . La solución óptima

$$x^* \in \operatorname{argmin}\{\mathbb{E}[f(x, \omega)] | x \in \cap_{\omega \in \Omega} \mathcal{C}^\omega\}, \tag{4.16}$$

representa la solución realística del problema de optimización estocástica. Esta solución, nos protege contra todas las posibles contingencias $\omega \in \Omega$ que pueden ocurrir en el futuro.

4.3.3. El enfoque en “Valor esperado”(Expected value)

Sea $\bar{\omega}$ la esperanza matemática (valor esperado) sobre el conjunto Ω , es decir:

$$\bar{\omega} = \mathbb{E}[\omega] = \sum_{\omega \in \Omega} \omega \mathbb{P}(\omega). \tag{4.17}$$

En el enfoque en valor esperado, reemplazamos los parámetros estocásticos por sus valores esperados y resolvemos el correspondiente problema determinístico:

$$\begin{aligned} \hat{z} &= \min f(x, \bar{\omega}) \\ \text{s.a} \\ x &\in \mathcal{C}^{\bar{\omega}}. \end{aligned} \tag{4.18}$$

Luego, la solución óptima será:

$$\hat{x} \in \operatorname{argmin}\{f(x, \bar{\omega}) | x \in \mathcal{C}^{\bar{\omega}}\}. \tag{4.19}$$

Nos referiremos al problema (4.18) como el “*problema de valor esperado*” correspondiente al problema de optimización estocástica (4.11) y \hat{z} denota el costo correspondiente a \hat{x} , quien viene a ser la solución óptima del “*expected value problem*”. Denotaremos por z_d el costo esperado correspondiente a la implementación de la solución del problema de valor esperado \hat{x} , es decir:

$$z_d = \mathbb{E}[f(\hat{x}, \omega)] \tag{4.20}$$

4.3.4. Evaluación y comparación de los diferentes enfoques

Asumiendo que $f(x, \omega)$ es convexa y comparando los costos esperados de las diferentes soluciones de los problemas de optimización estocástica que acabamos de ver, en [Infanger, 1992] se demuestra que

$$z_{ws} \leq z \leq z_d. \tag{4.21}$$

Definamos ahora el *valor esperado bajo información perfecta* (EVPI) como:

$$EVPI = z - z_{ws}. \quad (4.22)$$

Este es una medida de cuánto estaríamos dispuestos a pagar (como máximo) para obtener información perfecta sobre el futuro. Un pequeño EVPI indica que los pronósticos refinados conducirían a poca ganancia, mientras que un EVPI grande indica que la información incompleta sobre el futuro es costosa.

Análogamente, definamos también el *valor de la solución estocástica* (VSS) como:

$$VSS = z_d - z. \quad (4.23)$$

Este es una medida de cuánto se puede ahorrar, si implementamos la solución del problema de optimización estocástica en lugar de la solución del problema determinista de valor esperado. Si el VSS es pequeño, la aproximación del problema estocástico por el problema del valor esperado correspondiente es buena y la solución del valor esperado obtenido es una buena solución para el problema estocástico. Cuanto mayor sea el valor de VSS, más importante será obtener la solución del problema de optimización estocástica. Podemos encontrar un resultado interesante en ([Birge, 1980]), en donde se desarrollaron límites en el EVPI y en el VSS, en modo que:

$$\begin{aligned} 0 \leq EVPI &\leq z - \widehat{z} \leq z_d - \widehat{z}, \\ 0 \leq VSS &\leq z_d - \widehat{z}. \end{aligned} \quad (4.24)$$

Estos límites, son bastante útiles para decidir si es necesario resolver el problema estocástico o si una aproximación es adecuada.

Retomemos la discusión de nuestro modelo. En el PPMT que hemos planteado, nuestro horizonte temporal esta dado por T periodos indexados por $t = 1, \dots, T$, en donde las decisiones en cada periodo estan dadas por x_1, \dots, x_T respectivamente. Consideremos por ejemplo el instante $t = 1$ que es cuando iniciamos a ajecutar el modelo, entonces, es claro que existe mucha información que desconoceremos sobre lo que sucederá en el instante $t = 2$, y con mayor razón, sobre lo que sucederá en los instantes sucesivos a este último. Sin embargo, nosotros debemos de tomar la decisión ya (para $t = 1$), consecuentemente, este razonamiento cae dentro de la familia de decisiones “*here and now*”, en el cual, estamos obligados a determinar en el momento $t = 1$ ciertas decisiones x_1 que no pueden cambiar más adelante, pase lo que pase en los instantes posteriores. Esto conlleva a contemplar el escenario en que, una decisión en el momento inicial sea demasiado costosa en el futuro, y este efecto plantea la necesidad de un enfoque sistemático para tomar tales decisiones. Desarrollemos más en detalle estas ideas en la siguiente sección.

4.4. Modelo con Recursos

Los *módelos con recursos* son un tipo de modelos de optimización estocástica del tipo “*here an now*”, que consiste en optimizar las decisiones que debemos tomar en el instante $t = 1$ teniendo en cuenta las consecuencias de estas decisiones para todas las realizaciones de las aleatoriedades que puedan ocurrir más adelante. La palabra “*recurso*” revela la posibilidad de remediar una

decisión que puede ser demasiado optimista en el instante $t = 1$, mediante la implementación de acciones correctivas, obviamente más caras, que satisfarán las restricciones postuladas en los momentos posteriores $t = i, i > 1$. Debemos precisar que después de tomar la decisión en el instante $t = 1$, pasamos al instante $t = 2$ en donde se revelarán todos los eventos desconocidos hasta ese momento, de igual manera para el instante $t = 3, t = 4$, hasta el instante $t = T$. Por lo tanto, el estudio de tales casos es de interés solo si tenemos alguna información sobre la realización de eventos que desconocemos en el momento $t = 1$ (información como, por ejemplo las distribuciones de probabilidades, los escenarios, la media, la varianza, etc). Para entender mejor como funciona este modelo, iniciemos estudiando el *modelo con recursos en dos etapas*.

4.4.1. Modelo con recursos en dos etapas

Cuando consideramos solo dos etapas, el conjunto de decisiones es dividido en dos grupos:

- *First-stage decisions*: Se deben tomar aquí varias decisiones antes del experimento. Todas estas decisiones se denominan decisiones en la primera etapa y el período en que se toman estas decisiones se denomina *primera etapa*.
- *Second-stage decisions*: Se deben tomar aquí varias decisiones después del experimento. Todas estas decisiones se denominan decisiones en la segunda etapa y su correspondiente período se denomina *segunda etapa*.

Por lo general, las decisiones en la primera etapa son representadas por el vector x , las cuáles se consideran estructurantes, variables estratégicas que no pueden cambiar en la segunda etapa y cuyas decisiones deben tomarse dentro de las restricciones de esa etapa; mientras que las decisiones en la segunda etapa son representadas por el vector y o también $y(\omega)$ o incluso $y(\omega, x)$ si se desea enfatizar que las decisiones de la segunda etapa difieren en función del resultado del experimento aleatorio y de la decisión de la primera etapa y que pueden emprenderse para satisfacer restricciones que involucran variables de la primera etapa. La sucesión de eventos y decisiones puede ser resumido como

$$x \rightsquigarrow \xi(\omega) \rightsquigarrow y(\omega, x).$$

Todas estas ideas se resumen en el siguiente programa:

$$\underset{x \in \mathcal{X}}{\text{Minimizar}} \quad f(x) := \mathbb{E}[F(x, \omega)], \quad (4.25)$$

donde $F(x, \omega)$ es el valor óptimo del problema de la segunda etapa, el cuál esta dado por

$$\underset{y \in \mathcal{G}(x, \omega)}{\text{Minimizar}} \quad g(x, y, \omega). \quad (4.26)$$

Aquí, $\mathcal{X} \subseteq \mathbb{R}^n$, $g : \mathbb{R}^m \times \Omega \rightarrow \mathbb{R}$ y $\mathcal{G} : \mathbb{R}^n \times \Omega \rightrightarrows \mathbb{R}^m$ es una multifunción.

Este modelo un poco abstracto podemos entenderlo mejor analizando el caso lineal, el cuál es planteado del siguiente modo:

$$\begin{aligned} \underset{\substack{x \in \mathcal{R}^n \\ s.a.}}{\text{Minimizar}} \quad & c^T x + \mathbb{E}[Q(x, \xi)] \\ & Ax = b, \quad x \geq 0, \end{aligned} \quad (4.27)$$

donde, $Q(x, \xi)$ es el valor óptimo del problema de la segunda etapa

$$\begin{aligned} & \underset{y \in \mathcal{R}^m}{\text{Minimizar}} && q^\tau y \\ & \text{s.a.} && \\ & && Bx + Wy = h, \quad y \geq 0, \end{aligned} \tag{4.28}$$

Aquí, $\xi := (q, h, B, W)$ son los datos del problema de la segunda etapa en donde algunos o todos los elementos del vector pueden ser aleatorios y además, la esperanza matemática en la primera etapa (4.27) se toma con respecto a la distribución de probabilidad de ξ . A menudo, usaremos la misma notación ξ para denotar un vector aleatorio y una de sus realizaciones $\xi(\omega)$ en particular. En la siguiente sección generalizaremos las ideas abordadas en esta parte en el cuál el problema contempla más de dos etapas.

4.4.2. Modelo con recursos multietapas

Los modelos con recursos a dos etapas pueden ser extendidos de manera natural a uno multi-etapas, el cuál consiste en la optimización de un sistema en donde las decisiones deben tomarse sucesivamente en ciertas etapas de índices $t = 1, \dots, T$ y basarse en información adquirida progresivamente. De este modo, los datos inciertos ξ_1, \dots, ξ_T son revelados gradualmente en el tiempo durante los T períodos y, por lo tanto nuestras decisiones deben adaptarse a dicho proceso. Este proceso de decisión tiene la forma:

$$\text{decisión}(x_1) \rightsquigarrow \text{observ.}(\xi_1) \rightsquigarrow \text{decisión}(x_2) \rightsquigarrow \dots \rightsquigarrow \text{observ.}(\xi_{T-1}) \rightsquigarrow \text{decisión}(x_T).$$

Esto significa que ξ_{t-1} es observado antes de tomar la decisión x_t , este proceso de toma de decisiones puede observarse claramente en el cuadro 4.1:

Etapas	Información disponible para la toma de decisiones			Decisión
	Decisiones previas	Resultados observados	Incertidumbre residual	
1	ninguna	ninguna	$\mathbb{P}[(\xi_1, \dots, \xi_T)]$	x_1
2	x_1	ξ_1	$\mathbb{P}[(\xi_2, \dots, \xi_T) \xi_1]$	x_2
3	x_1, x_2	ξ_1, ξ_2	$\mathbb{P}[(\xi_3, \dots, \xi_T) \xi_1, \xi_2]$	x_3
\vdots				\vdots
T	x_1, x_2, \dots, x_{T-1}	$\xi_1, \xi_2, \dots, \xi_{T-1}$	$\mathbb{P}[\xi_T \xi_1, \xi_2, \dots, \xi_{T-1}]$	x_T
opcional: $T + 1$	x_1, x_2, \dots, x_T	$\xi_1, \xi_2, \dots, \xi_T$	ninguna	x_{T+1}

Tabla 4.1: Etapas de decisión, estableciendo el orden de las observaciones y decisiones.

Como regla general, para modelar la dependencia de una decisión sobre cierta información, procedemos de la siguiente manera:

- (1) Modelizamos la información en cuestión con la ayuda de un sub σ -álgebra $\mathcal{G} \subseteq \mathcal{F}$ que define los eventos de Ω y los cuáles son distinguibles para la toma de decisiones.
- (2) Expresamos la decisión como una función denominada “a policy”³ definida en $\Omega \rightarrow \mathbb{R}^n$, es decir, un vector aleatorio.

³Esta función también es conocida como *estrategia* en el campo de la economía.

(3) Forzamos a que esta función sea medible respecto a \mathcal{G} .

En nuestro caso, la información accesible en la etapa t es modelizado por un σ -álgebra \mathcal{F}_t (completado con los conjuntos de medida nula) generado por el vector:

$$\pi_t = (\xi_1, \dots, \xi_t) \quad (4.29)$$

que representa el pasado del escenario en el período t , es decir, las incertezas que uno habrá observado y sobre el cuál nos basaremos para tomar la decisión x_t . La sucesión de σ -álgebras

$$\mathcal{F}_1 \subseteq \mathcal{F}_2 \subseteq \dots \subseteq \mathcal{F}_T \subseteq \mathcal{F}$$

define de este modo una *filtración*⁴, en donde esta sucesión de inclusiones significa que la adquisición de la información es progresiva y sin pérdida de memoria.

Denotaremos de aquí en adelante con $X_t \in L^\infty(\Omega, \mathcal{F}, \mathbb{P}, \mathbb{R}^n)$ la decisión tomada en la etapa t y el cuál debe ser \mathcal{F}_t -medible. Estas restricciones de medibilidad son conocidas como restricciones de *no-anticipatividad*: es decir, X_t no se anticipa al futuro visto desde el período t , en cuyo caso, existirá una función medible φ_t que verifique:

$$X_t = \varphi_t \circ \pi_t \quad (4.30)$$

explicando la dependencia de la decisión sobre la información. Finalmente, el problema de minimización multietapa con recurso se puede escribir como:

$$\begin{aligned} & \underset{(X_1, \dots, X_T)}{\text{Minimizar}} && \mathbb{E}[f(X_1, \dots, X_T; \xi_1, \dots, \xi_T)] \\ & s.a. && X_1, \dots, X_T \in L^\infty(\Omega, \mathcal{F}, \mathbb{P}, \mathbb{R}^n) \\ & && X_t \text{ es } \mathcal{F}_t\text{-medible} \end{aligned} \quad (4.31)$$

Es evidente que este problema no puede ser abordado numéricamente en forma directa, ya que manipula en general objetos de dimensión infinita (las funciones). Por lo tanto, este solo puede resolverse aproximadamente; además, en muchas aplicaciones no siempre conocemos la ley de ξ sino solo un conjunto de escenarios $\{\xi^{(1)}, \dots, \xi^{(S)}\}$ proveniente de resultados experimentales o registros históricos. Una forma de abordar el problema es reemplazar ξ por una variable aleatoria discreta

$$\hat{\xi} = (\hat{\xi}_1, \dots, \hat{\xi}_T)$$

tomando el valor ξ^s con probabilidad p_s ; $\sum_{s=1}^S p_s = 1$, en modo similar a como hicimos para el caso a dos etapas. Así, podemos definir el análogo de π_t definido en (4.29), por $\hat{\pi}_t$ que describe el pasado de $\hat{\xi}$, es decir:

$$\hat{\pi}_t = (\hat{\xi}_1, \dots, \hat{\xi}_t)$$

que también es discreta. Este, toma sus valores en el conjunto $\{\pi_t^{(1)}, \dots, \pi_t^{(S)}\}$ con $\pi_t^{(s)} = (\xi_1^{(s)}, \dots, \xi_t^{(s)})$ por construcción. El proceso $\hat{\xi} = (\hat{\xi}_1, \dots, \hat{\xi}_T)$ genera de este modo también una filtración

⁴Ver apéndice B

$$\hat{\mathcal{F}}_1 \subseteq \hat{\mathcal{F}}_2 \subseteq \dots \subseteq \hat{\mathcal{F}}_T \subseteq \mathcal{F}$$

que podemos emplear para expresar la *no-anticipatividad*. El problema aproximado de 4.31 es entonces

$$\begin{aligned} & \underset{(X_1, \dots, X_T)}{\text{Minimizar}} && \mathbb{E} [f(X_1, \dots, X_T; \hat{\xi}_1, \dots, \hat{\xi}_T)] \\ & \text{s.a.} && \\ & && X_1, \dots, X_T \in L^\infty(\Omega, \mathcal{F}, \mathbb{P}, \mathbb{R}^n) \\ & && X_t \text{ es } \hat{\mathcal{F}}_t - \text{medible} \end{aligned} \tag{4.32}$$

La naturaleza discreta del proceso $\hat{\xi}$ permite representar gráficamente la evolución de las incertezas sobre el tiempo como la forma de un *árbol de escenarios* como el de la Figura 4.2. Un árbol de escenarios es una representación explícita del proceso de ramificación inducido por la observación gradual de ξ_1, \dots, ξ_T , bajo el supuesto de que las variables aleatorias tienen un soporte discreto⁵. Está construido de la siguiente manera: a la primera etapa de decisión viene asociado un *nodo raíz* con ausencia inicial de observaciones. Al nodo raíz, vienen conectados los nodos secundarios que corresponden a la etapa 2, un nodo secundario para cada posible resultado de la variable aleatoria ξ_1 . Luego, a cada nodo de la etapa 2 se conectan nodos secundarios asociados a la etapa 3, uno para cada resultado de ξ_2 condicionada a la observación de ξ_1 relativa al nodo padre. La construcción del proceso de ramificación continúa hasta que se alcanza la última etapa; en este punto, los resultados asociados a los nodos en la ruta única desde la raíz a una hoja definen juntos un escenario particular que puede asociarse a la hoja.

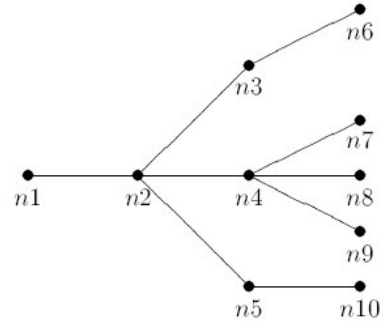


Figura 4.2: Árbol de escenarios con 5 escenarios y 10 nodos.

Para visualizar cómo la información sobre la realización de las variables aleatorias se vuelve gradualmente disponible, es conveniente imaginar particiones anidadas del espacio de eventos (Figura 4.3): los refinamientos de las particiones aparecen gradualmente en cada etapa de decisión en correspondencia con las posibles realizaciones de la nueva observaciones. A cada subregión inducida por la partición del espacio de eventos se le puede asociar una decisión de recurso constante, como si las decisiones se eligieran de acuerdo con una política de decisión constante por partes.

Idealmente, un árbol de escenarios debería cubrir la totalidad de los posibles resultados de un proceso aleatorio. Pero a menos que el soporte de la distribución de las variables aleatorias sea finito, ningún árbol de escenarios con un número finito de nodos puede representar exactamente el proceso aleatorio y la medida de probabilidad, incluso si el soporte es finito, el número de Los escenarios crecen exponencialmente con el número de etapas. Sin embargo, la aproximación del árbol de escenarios finitos es necesaria porque los métodos de optimización numérica no pueden manejar directamente problemas en espacios de dimensión infinita como mencionamos anteriormente.

⁵Es el conjunto cerrado más pequeño Ξ en \mathbb{R}^T tal que $\mathbb{P}[(\xi_1, \dots, \xi_T) \in \Xi] = 1$, también llamado soporte de la distribución conjunta.

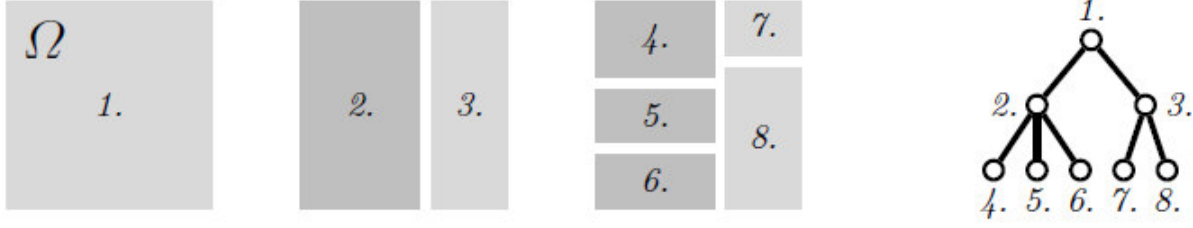


Figura 4.3: (De izquierda a derecha) Partición anidada del espacio de eventos Ω , comenzando desde una partición trivial que representa la ausencia de observaciones. (El extremo derecho) Árbol de escenarios correspondiente al proceso de partición.

La aproximación puede servir para proporcionar una estimación del valor óptimo del programa original, también puede servir para obtener una decisión x_1 aproximada de la primera etapa.

Para fijar ideas, ilustremos la técnica del árbol de escenarios sobre un problema de seguimiento de una trayectoria bajo incertidumbre con control de penalización. En el ejemplo propuesto, la incertidumbre es tal que el problema exacto se puede plantear en un pequeño árbol de escenarios finitos. Consideremos entonces la evolución de un sistema dinámico con estados continuos y en tiempo discreto de la forma

$$x_{t+1} = x_t + v_t + \xi_t$$

donde $x_t \in \mathbb{R}$ es el estado del sistema dinámico, t es el tiempo, v_t es el control de la penalización y ξ_t es la perturbación en la evolución del sistema dinámico.

	$k :=$	1	2	3	4	5	6	7
$t = 1$	ξ_1^k	-4	-4	-4	3	3	3	3
$t = 2$	ξ_2^k	-3	2	2	-3	0	0	2
$t = 3$	ξ_3^k	0	-2	1	0	-1	2	1
p	p^k	0.1	0.2	0.1	0.2	0.1	0.1	0.2

Tabla 4.2: Distribución de probabilidad de la perturbación aleatoria ξ .

Por simplicidad, consideraremos un período $T = 3$, por lo tanto $t = 1, 2, 3$ y además supongamos que el sistema inicia en el estado $x_1 = 0$. Supongamos además que conocemos la distribución de probabilidad de la perturbación aleatoria $\xi = (\xi_1, \xi_2, \xi_3)$ el cual es mostrado en el cuadro 4.2. Observamos rápidamente que para los tiempos $t = 1, 2, 3$ tenemos 7 posibles resultados para el proceso aleatorio, los cuales son denotados por ξ^k , $1 \leq k \leq 7$ con probabilidades p^k . Este proceso estocástico puede ser completamente representado por un árbol de escenarios como se muestra en la Figura 4.4. De aquí, se puede ver rápidamente que por ejemplo el

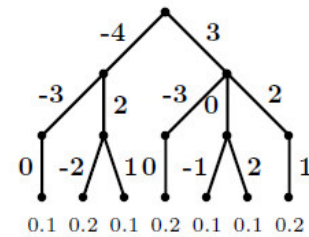


Figura 4.4: Árbol que representa los 7 escenarios posibles para un proceso aleatorio $\xi = (\xi_1, \xi_2, \xi_3)$. Los resultados ξ_t^k están en negrita y las probabilidades p^k de cada escenario se reportan en los nodos de cada hoja.

primer posible resultado es $\xi^1 = (-4, -3, 0)$ y tiene una probabilidad $p^1 = 0,1$, además, observemos también que las variables aleatorias ξ_1, ξ_2, ξ_3 no son mutuamente independientes.

Supongamos ahora que un agente puede elegir las acciones $v_t \in \mathbb{R}$ para cada $t = 1, 2, 3$. El objetivo del agente será obviamente minimizar la esperanza de la suma de los costos $C_t(v_t, x_{t+1})$ condicionada al estado inicial del sistema, es decir, matemáticamente:

$$\text{Minimizar } \mathbb{E} \left[\sum_{t=1}^3 C_t(v_t, x_{t+1}) \mid x_1 = 0 \right].$$

Los costos $C_t(v_t, x_{t+1})$, asociados a la decisión v_t y la transición al estado x_{t+1} , se definen por $C_t = \left(d_{t+1} + \frac{v_t^2}{4} \right)$ con $d_{t+1} = |x_{t+1} - \alpha_{t+1}|$ y $\alpha_2 = 2,9$, $\alpha_3 = 0$, $\alpha_4 = 0$. Aquí, α_{t+1} es la trayectoria nominal elegida arbitrariamente; d_{t+1} es el error de seguimiento y $\frac{v_t^2}{4}$ es la penalización del esfuerzo de control.

Una *policy* óptima que mapea las observaciones ξ_1, \dots, ξ_{t-1} a las decisiones v_t se puede obtener resolviendo el programa cuadrático convexo (4.33) definido sobre las variables $v_t^k, x_{t+1}^k, d_{t+1}^k$, donde k varía de 1 a 7 y t de 1 a 3, y además, trivialmente tenemos $x_1^k = 0$:

$$\begin{aligned} \text{Minimizar} \quad & \sum_{k=1}^7 p^k \left[\sum_{t=1}^3 \left(d_{t+1}^k + \frac{(v_t^k)^2}{4} \right) \right] \\ \text{s.a.} \quad & -d_{t+1}^k \leq x_{t+1}^k - \alpha_{t+1} \leq d_{t+1}^k, \forall k, t \\ & x_1^k = 0, \quad x_{t+1}^k = x_t^k + v_t^k + \xi_t^k, \forall k, t \\ & v_1^1 = v_1^2 = v_1^3 = v_1^4 = v_1^5 = v_1^6 = v_1^7 \\ & v_2^1 = v_2^2 = v_2^3, \quad v_2^4 = v_2^5 = v_2^6 = v_2^7 \\ & v_3^2 = v_3^3, \quad v_3^5 = v_3^6. \end{aligned} \tag{4.33}$$

Este problema puede ser resuelto fácilmente con un software como por ejemplo Matlab (con el paquete “CVX”). Después de implementar el algoritmo, el código en efecto devuelve la solución óptima correspondiente a las variables implicadas en este problema, el cuál se muestra en la Figura 4.5, de donde reemplazando estos valores en la función objetivo obtenemos el valor óptimo +7,3148. En este ejemplo, observe que la solución final se puede emitir como mapeos π_t de x_t hacia v_t , es decir, $\pi_1(0) = -1$, $\pi_2(-4,1) = 2,1$, $\pi_2(2,9) = -1,16$, $\pi_3(-5) = 2$, $\pi_3(-1,26) = 1,26$, $\pi_3(0) = 0,667$, $\pi_3(1,74) = -0,74$, $\pi_3(3,74) = -2$.

De manera análoga, gracias a un proceso de discretización, el problema 4.32 es, de hecho, un

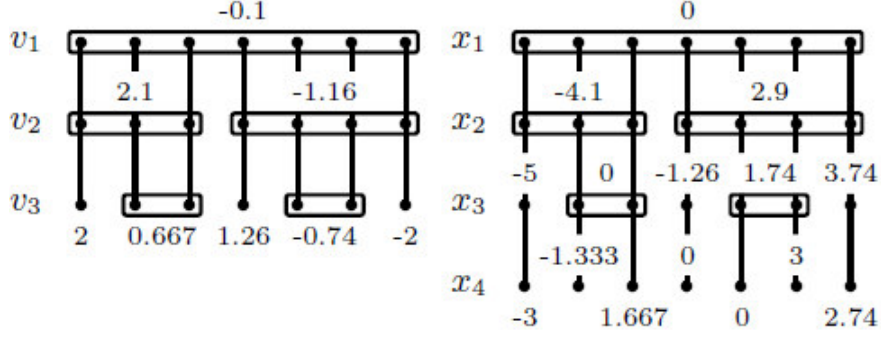


Figura 4.5: (Izquierda) Se muestra las acciones óptimas v_t para el agente escenario por escenario, con marcos alrededor de los escenarios que pasan por un mismo nodo de árbol. (Derecha) Estados visitados x_t bajo las acciones óptimas, tratadas como decisiones artificiales.

problema en dimensión finita:

$$\begin{aligned}
 & \underset{x_t^s}{\text{Minimizar}} && \sum_{s=1}^S p_s f(x^{(s)}; \xi^{(s)}) \\
 & \text{s.a.} && x^{(s)} = x^{(s')} \forall (s, s', t) / \pi_t^{(s)} = \pi_t^{(s')}
 \end{aligned} \tag{4.34}$$

donde

$$\begin{pmatrix} x_t^{(1)} \\ \vdots \\ x_t^{(S)} \end{pmatrix} = \begin{pmatrix} \varphi_t(\pi_t^{(1)}) \\ \vdots \\ \varphi_t(\pi_t^{(S)}) \end{pmatrix} \in \mathbb{R}^{nS} \tag{4.35}$$

y la función φ_t es aquella definida en 4.30. La *no-anticipatividad* se expresa aquí de la siguiente manera: si en el período t , no se puede distinguir entre dos escenarios (s) y (s'), es decir $\pi_t^{(s)} = \pi_t^{(s')}$, entonces se debe tomar la misma decisión para estos dos escenarios:

$$\pi_t^{(s)} = \pi_t^{(s')} \Rightarrow x_t^{(s)} = x_t^{(s')}.$$

Esto supone implícitamente que la discretización de las variables aleatorias se realiza de tal manera que todo el universo de posibles eventos puede ser descrito por este conjunto de escenarios.

Aquí hay dos frenos para el uso de los árboles. Si solo tenemos escenarios, es decir, no tenemos un mecanismo para simular las leyes condicionales del proceso ξ , se debe hacer mucho esfuerzo para transformar un paquete de escenarios en un árbol. En segundo lugar, cuando se trata de un programa estocástico de múltiples etapas, para tomar las decisiones a menudo estamos interesados no solo en la primera decisión, sino también en las primeras decisiones de los recursos. Desafortunadamente, la topología de los árboles no suele proporcionar suficientes nodos en la primera etapa para permitir la construcción de una aproximación funcional (una discusión de este tema es tratado en [Daldoul, 1997] y [Dallagi, 2007]). En [Lenoir, 2008], se proponen dos métodos que a diferencia de los árboles de escenarios, estos no requieren una modificación de la muestra de datos, sino que lo mantienen tal como está. Las restricciones de la no-anticipación en este contexto, se abordan a través de un esquema de discretización basado en las identidades del estimador de regresión no paramétrica de Nadaraya-Watson (ver [Nadaraya, 1964; Watson, 1964]), temas que escapan a

los objetivos de esta tesis. Por consiguiente, nosotros recogeremos los resultados desarrollados en profundidad en [Lenoir, 2008], en donde se demuestra que los tres acoplamientos naturales del problema de la gestión de la producción de energía eléctrica en entornos aleatorios, a saber:

- (1) *en espacio*: las decisiones sobre las distintas plantas generadoras están vinculadas a tener que satisfacer la demanda.
- (2) *en tiempo*: las decisiones sobre los distintos pasos de tiempo están vinculadas por las restricciones dinámicas de funcionamiento de las plantas generadoras.
- (3) *en incertezas*: las decisiones sobre los diferentes escenarios están vinculadas por la necesidad de no anticipar el futuro,

presenta la estructura:

$$\begin{aligned}
 \underset{p_{u,t}^s}{\text{Minimizar}} \quad & \sum_{u,t,s} f_{u,t}^{(s)}(p_{u,t}^{(s)}) + \\
 & \sum_{u,t} C_{u,t}(p_{u,t}^{(1)}, \dots, p_{u,t}^{(s)}) + \\
 & \sum_{t,s} C_t^{(s)}(p_{1,t}^{(s)}, \dots, p_{U,t}^{(s)}) + \\
 & \sum_{u,s} C_u^{(s)}(p_{u,1}^{(s)}, \dots, p_{u,T}^{(s)})
 \end{aligned} \tag{4.36}$$

donde $f_{u,t}^{(s)}$ representa el costo de la planta u en el período t del escenario s ; $C_{u,t}(\cdot)$ es el acoplamiento en incertezas, $C_t^{(s)}(\cdot)$ el acoplamiento espacial y $C_u^{(s)}(\cdot)$ el acoplamiento temporal. En el siguiente capítulo veremos dos aplicaciones sencillas del tipo 4.36 junto con su solución numérica para entender la estructura que se esconde detrás de estos conceptos.

Capítulo 5

Aplicación: La gestión de la producción de energía eléctrica

Como mencionamos en el capítulo 4, existen diversas formas de producir energía eléctrica (E.E.): termonuclear, térmica, eólica, hidráulica, etc. En el Perú, contamos con 5 diferentes tipos de tecnología para la producción de energía eléctrica, la termoeléctrica, la hidroeléctrica, la eólica, la fotovoltaica y la generación por biomasa. En este capítulo, presentaremos dos modelos sencillos para la gestión de la producción eléctrica considerando solo funciones cuadráticas de costo. El primer modelo, tiene como objetivo mostrar como introducir los efectos aleatorios en el análisis de la gestión de la producción eléctrica, sin embargo, solo consideraremos dos posibles escenarios y 3 períodos de tiempo para nuestra experimentación. En el segundo modelo aumentaremos la cantidad de períodos hasta 240 y 1488, no obstante, para este caso solo tomaremos en cuenta un posible escenario (lo cuál se reduce al caso determinístico), considerando N_{hid} centrales hidroeléctricas, N_{ter} centrales termoeléctricas y N_{des} de un tipo de central arbitraria, donde $N_{hid}, N_{ter}, N_{des} \in \mathbb{Z}^+$. Las restricciones que acabamos de plantear en ambos modelos, se fundamentan principalmente en la capacidad de procesamiento de datos, pues todas nuestras experimentaciones la estamos realizando solo con un CPU de escritorio. Por otro lado, recordemos que el funcionamiento de los tipos de tecnologías para la generación eléctrica es por supuesto completamente diferente. Por ejemplo, las centrales térmicas pueden ser cerradas, es decir, el costo de operación de una planta de energía térmica es proporcional a su tiempo de operación, a diferencia de las plantas de energía hidráulica que operan “gratis”. Las plantas de energía hidráulica tienen un stock de energía, que es una reserva que puede ser suministrada (lluvia, deshielos, etc.). En consecuencia, la dificultad de administrar estas plantas no se encuentra en el nivel de los costos operativos, sino en el nivel de la administración: es necesario dividir su uso en todo el horizonte considerado para no agotar las reservas. También debe tenerse en cuenta que su funcionamiento es reversible, es decir, pueden consumir energía para aumentar su stock. Veamos esto un poco más en detalle.

5.1. Descripción de las diferentes tecnologías de producción de E.E. en el Perú

Recordemos que nuestro plan de producción es a mediano plazo (PPMT), por lo tanto disponemos de T periodos indexados por $t = 1, \dots, T$ con duraciones respectivas de $\delta_1, \dots, \delta_T$. Además, disponemos de una cantidad finita de plantas de producción eléctrica indexadas por las letras

$u = 1, \dots, U$, cada una de las cuáles genera durante cada periodo t una potencia $p_{u,t} \in \mathbb{R}$. El costo instantáneo de esta producción esta representada por la función $f_{u,t} : \mathbb{R} \rightarrow \mathbb{R}$. También, existe un costo C_u el cuál dependerá de la estrategia de producción sobre todo el horizonte temporal $(p_{u,1}, p_{u,2}, \dots, p_{u,T}) \in \mathbb{R}^T$.

5.1.1. Centrales Termoeléctricas

Una central termoeléctrica es una instalación empleada en la generación de energía eléctrica a partir de la energía liberada por combustibles fósiles como el petróleo, gas natural, carbón y núcleos de uranio. Este calor es empleado por un ciclo termodinámico convencional para mover un alternador y producir energía eléctrica, liberando dióxido de carbono a la atmósfera. Este tipo de plantas presentan las siguientes características:

- **Restricciones de funcionamiento:** Existen numerosas restricciones en la operación de estas plantas, como por ejemplo, restricciones de tiempo de funcionamiento mínimo en niveles de potencia fijos, tiempo mínimo de encendido o apagado, número máximo de caídas de potencia o arranques en periodos determinados, etc.
- **Costos de producción:** Los costos de producción de una planta termoeléctrica se descompone en tres partes:
 - Un costo de puesta en marcha en función de la duración de la parada anterior.
 - Un costo fijo independiente de la potencia generada.
 - Un costo proporcional a la potencia generada.

Estos costos están relacionados con los precios del combustible y las cuotas de CO2 restantes que se utilizarán. También varían según las características de cada unidad termoeléctrica. Además, existen otros costos relacionados con el mantenimiento y la mano de obra.

De acuerdo a estas características tenemos:

$$0 \leq p_{u,t} \leq \bar{P}_u, \quad \forall t = 1, \dots, T.$$

Como primera aproximación, el costo operativo es lineal en función de la producción e independiente del tiempo. Por lo tanto, la función de costo instantáneo asociada será:

$$f_{u,t} = \begin{cases} c_u p_{u,t}, & \text{si } p_u \in [0, \bar{P}_u] \\ +\infty, & \text{otro caso.} \end{cases}$$

Además, las decisiones son independientes del tiempo, de donde tenemos $C_u \equiv 0$. Debemos señalar sin embargo, que existen muchas situaciones en donde se tiene que $f_{u,t} = ap_{u,t}^2 + bp_{u,t} + c$, donde a, b, c son constantes.

5.1.2. Centrales Hidroeléctricas

Una central hidroeléctrica utiliza la energía cinética del agua contenida en un reservorio, un río en general, y la transforma en energía eléctrica a través de un alternador. Las ventajas de las centrales hidroeléctricas son:

- No necesitan combustibles y son limpias.
- Muchas veces los embalses de las centrales tienen otras utilidades importantes: regadío, como protección contra las inundaciones o para suministrar agua a las poblaciones próximas.
- Tienen costes de explotación y mantenimientos bajos.
- Las turbinas hidráulicas son de fácil control y tienen unos costes de mantenimiento reducido.

Para poder modelar matemáticamente este tipo de plantas eléctricas, consideremos que el nivel inicial del reservorio esta dado por $\ell_{u,0}$, este varía con el pasar del tiempo y por lo tanto su nivel al final del período $t = 1, \dots, T$ estará dado por $\ell_{u,t}$. Al inicio de cada período t , existirá un aporte natural $a_{u,t}$ debido a precipitaciones, lluvias, deshielos, etc; que alimentarán el reservorio. Entonces, existen dos decisiones que debemos considerar: ya sea para vaciarlo parcialmente con el objetivo de impulsar el alternador o simétricamente, para bombear el agua con el fin de reponer el stock consumiendo esta vez electricidad.

Si denotamos por $v_{u,t}$ el volumen de agua que transita de arriba hacia abajo (río arriba a río abajo), el flujo a través de la planta será $v_{u,t}/\delta_t$. Esto genera una potencia suministrada durante tal período (siempre denotado por $p_{u,t}$) gracias a una función de rendimiento $r_u : \mathbb{R} \rightarrow \mathbb{R}$ equivalente a:

$$p_{u,t} = r_u(v_{u,t}/\delta_t). \quad (5.1)$$

Esta potencia será positiva en caso de turbina y negativa en caso de bombeo; en el segundo caso, tendremos consumo de energía en lugar de producción. Al igual que con las centrales termoeléctricas, la potencia está limitada de manera inferior y superpuesta dentro de un rango $[\underline{P}_u, \bar{P}_u]$. De aquí, el nivel del reservorio estará limitada a las capacidades máximas y mínimas:

$$\underline{L}_u \leq \ell_{u,t} \leq \bar{L}_u. \quad (5.2)$$

Para respetar esta restricción, debemos considerar la posibilidad, en ciertos casos, de desechar un eventual excedente de agua sin utilizarla en producción. Así, la ecuación dinámica del reservorio se escribe:

$$\ell_{u,t} = \min\{\ell_{u,t-1} + a_{u,t} - v_{u,t}, \bar{L}_u\}. \quad (5.3)$$

Por lo tanto, el costo de funcionamiento de las plantas hidroeléctricas dependen de la altura final del reservorio, el cuál puede ser valorizado con la ayuda de la función $\mathcal{V}_u : \mathbb{R} \rightarrow \mathbb{R}$. Esta altura final depende del conjunto de las producciones sobre el horizonte temporal, el cuál tiene la forma:

$$C(p_{u,1}, \dots, p_{u,T}) = \begin{cases} -\mathcal{V}_u(\ell_{u,T}), & \text{si } (5.1), (5.2), (5.3); \\ +\infty, & \text{otro caso.} \end{cases} \quad (5.4)$$

El costo instantaneo es independiente de la cantidad de agua utilizada y consecuentemente de la potencia producida, de aquí tenemos:

$$f_{u,t}(p_{u,t}) = \begin{cases} 0, & \text{si } p_{u,t} \in [\underline{P}_u, \bar{P}_u]; \\ +\infty, & \text{otro caso.} \end{cases} \quad (5.5)$$

5.1.3. Centrales de Generación con Recursos Energéticos Renovables (RER)

Central Solar Fotovoltaica

La energía solar fotovoltaica se aprovecha transformándola directamente en electricidad mediante el efecto fotovoltaico. En la instalación fotovoltaica la obtención de energía eléctrica se produce a través de paneles fotovoltaicos que captan la energía luminosa del Sol para transformarla en energía eléctrica. Para conseguir la transformación se emplean células fotovoltaicas fabricadas con materiales semiconductores. Los paneles fotovoltaicos pueden ser del tipo fijo o móviles a través de seguidores solares.

Centrales de Generación Biomasa

La energía de la biomasa proviene en última instancia del Sol. Los vegetales absorben y almacenan una parte de la energía solar que llega a la tierra y a los animales, en forma de alimento y energía. Cuando la materia orgánica almacena la energía solar, también crea subproductos que no sirven para los animales ni para fabricar alimentos pero sí para hacer energía de ellos. Mediante métodos termoquímicos se transforma la energía biomasa utilizando el calor, estos pueden ser: combustión, pirólisis y gasificación. Una central de biomasa es una instalación industrial diseñada para generar energía eléctrica a partir de recursos biológicos (combustible metano de la descomposición orgánica). Así pues, las centrales de biomasa utilizan fuentes renovables para la producción de energía eléctrica.

Centrales de Generación Eólica

La energía eólica es la energía obtenida del viento, este es un efecto derivado del calentamiento desigual de la superficie de la Tierra por el Sol. La energía es utilizada principalmente para producir energía eléctrica. La energía eólica es un recurso abundante, renovable, limpio y ayuda a disminuir las emisiones de gases de efecto invernadero al reemplazar termoeléctricas a base de combustibles fósiles, lo que la convierte en un tipo de energía verde. En el parque eólico la producción de la energía eléctrica se consigue a partir de la fuerza del viento, mediante aerogeneradores que aprovechan las corrientes de aire. El principal problema de los parques eólicos es la incertidumbre respecto a la disponibilidad de viento cuando se necesita. Lo que implica que la energía eólica no puede ser utilizada como fuente de energía única y deba estar respaldada siempre por otras fuentes de energéticas con mayor capacidad de regulación (térmicas, nucleares, hidroeléctricas, etc.). La energía eólica requiere condiciones de intensidad y regularidad en el régimen de vientos para poder aprovecharlos. Se considera que vientos con velocidades promedio entre 5 y 12,5 m/s son los aprovechables.

5.2. Un primer modelo para la gestión de la producción de E.E.

A continuación presentamos un modelo sencillo para un caso particular de dos centrales de generación eléctrica ($u = 1, 2$.) cuyas funciones de costo están dadas por las funciones cuadráticas convexas $f_{1,t}(p_{1,t}) = 2p_{1,t}^2 - 2p_{1,t} + 3$ y $f_{2,t}(p_{2,t}) = 1,5p_{2,t}^2 - p_{2,t} + 2$. Además consideraremos un PPMT

con $T = 3$ (3 días), por lo tanto $t = 0, 1, 2, 3$ con $\delta_t = 1$ para $t = 1, 2, 3$. Para este caso en particular tendríamos el problema de optimización

$$\begin{aligned}
 & \underset{\substack{p_{u,t} \\ s.a.}}{\text{Minimizar}} && \sum_{t=0}^3 f_{1,t}(p_{1,t}) + f_{2,t}(p_{2,t}) \\
 & && \underline{P}_{1,t} \leq p_{1,t} \leq \bar{P}_{1,t}, \quad \forall t = 0, \dots, 3, \\
 & && \underline{P}_{2,t} \leq p_{2,t} \leq \bar{P}_{2,t}, \quad \forall t = 0, \dots, 3.
 \end{aligned} \tag{5.6}$$

Donde $\underline{P}_{1,t}$, $\underline{P}_{2,t}$ y $\bar{P}_{1,t}$, $\bar{P}_{2,t}$ son los límites mínimos y máximos de producción en el período t para las potencias de las centrales de generación eléctrica 1 y 2 respectivamente. Por simplicidad, consideraremos $\underline{P}_{1,t} = \underline{P}_{2,t} = 0$ y $\bar{P}_{1,t} = \bar{P}_{2,t} = 10$ para todo $t = 0, 1, 2, 3$. En forma matricial el problema 5.6 puede ser expresado como

$$\begin{aligned}
 & \underset{\substack{x_t \\ s.a.}}{\text{Minimizar}} && \sum_{t=0}^3 0,5x_t^T A x_t + b x_t + c \\
 & && \underline{P}_t \leq x_t \leq \bar{P}_t, \quad \forall t = 0, \dots, 3,
 \end{aligned} \tag{5.7}$$

donde

$$A = \begin{pmatrix} 4 & 0 \\ 0 & 3 \end{pmatrix}, \quad b = (-2, -1), \quad c = 5,$$

además $x_t = (p_{1,t}, p_{2,t})^T$ representa la variable de decisión para cada período t , $\underline{P}_t = (0, 0)^T$ y $\bar{P}_t = (10, 10)^T$ son los límites inferiores y superiores respectivamente para cada período. Los problemas 5.6 y 5.7 modelan la producción de energía eléctrica durante un período de tres días en donde los parámetros de las funciones de costo y las restricciones son valores determinísticos fijos para cada período $t = 0, 1, 2, 3$, los cuáles pueden ser resueltos fácilmente por los métodos clásicos de optimización no lineal.

Sin embargo, esto no siempre es un supuesto realista. Puede suceder que al menos algunos de los datos pueden variar dentro de ciertos límites (para nuestra discusión, aleatoriamente) y que tengamos que tomar una decisión sobre el plan de producción antes de conocer los valores exactos de esos datos.

A manera de ejemplo, supongamos que la matriz A puede variar de dos formas diferentes para cada período $t = 1, 2, 3$ con igual probabilidad para ambos casos. Luego, nuestro espacio muestral tendría la siguiente forma $\Omega = \{a, b\}$. Si quisieramos modelar estas variaciones, podemos apoyarnos de un diseño de árbol de escenarios como se muestra en la figura 5.1, donde

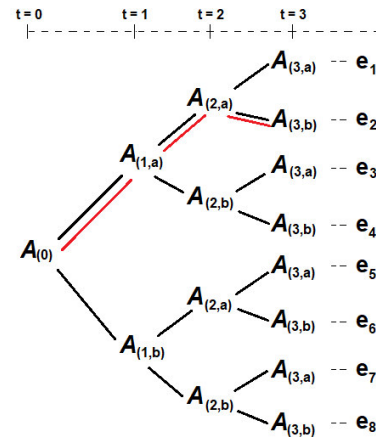


Figura 5.1: Árbol de escenarios para dos resultados posibles con $T = 3$ periodos.

la matriz $A_{(0)} = A$ y las matrices $A_{t,a}$, $A_{t,b}$ representan las dos posibles variaciones en cada período t cuando $t = 1, 2, 3$, las cuáles estan dadas por:

$$A_{(0)} = \begin{pmatrix} 4 & 0 \\ 0 & 3 \end{pmatrix},$$

$$A_{(1,a)} = \begin{pmatrix} 3,5 & 0 \\ 0 & 3,5 \end{pmatrix}, \quad A_{(1,b)} = \begin{pmatrix} 5 & 0 \\ 0 & 2 \end{pmatrix},$$

$$A_{(2,a)} = \begin{pmatrix} 4,1 & 0 \\ 0 & 2 \end{pmatrix}, \quad A_{(2,b)} = \begin{pmatrix} 3,8 & 0 \\ 0 & 4,8 \end{pmatrix},$$

$$A_{(3,a)} = \begin{pmatrix} 3,5 & 0 \\ 0 & 3,7 \end{pmatrix}, \quad A_{(3,b)} = \begin{pmatrix} 5,5 & 0 \\ 0 & 2,5 \end{pmatrix},$$

El problema de optimización si consideramos estas variaciones aleatorias sería entonces:

$$\begin{aligned} \underset{y_{t,e}}{Min} \quad & \frac{1}{8} \left\{ \begin{aligned} & 0,5y_{0,1}^T A_{(0)} y_{0,1} + by_{0,1} + c + 0,5y_{1,1}^T A_{(1,a)} y_{1,1} + by_{1,1} + c \\ & + 0,5y_{2,1}^T A_{(2,a)} y_{2,1} + by_{2,1} + c + 0,5y_{3,1}^T A_{(3,a)} y_{3,1} + by_{3,1} + c \end{aligned} \right. \\ & + \left[\begin{aligned} & 0,5y_{0,2}^T A_{(0)} y_{0,2} + by_{0,2} + c + 0,5y_{1,2}^T A_{(1,a)} y_{1,2} + by_{1,2} + c \\ & + 0,5y_{2,2}^T A_{(2,a)} y_{2,2} + by_{2,2} + c + 0,5y_{3,2}^T A_{(3,b)} y_{3,2} + by_{3,2} + c \end{aligned} \right] \\ & + \left[\begin{aligned} & 0,5y_{0,3}^T A_{(0)} y_{0,3} + by_{0,3} + c + 0,5y_{1,3}^T A_{(1,a)} y_{1,3} + by_{1,3} + c \\ & + 0,5y_{2,3}^T A_{(2,b)} y_{2,3} + by_{2,3} + c + 0,5y_{3,3}^T A_{(3,a)} y_{3,3} + by_{3,3} + c \end{aligned} \right] \\ & + \left[\begin{aligned} & 0,5y_{0,4}^T A_{(0)} y_{0,4} + by_{0,4} + c + 0,5y_{1,4}^T A_{(1,a)} y_{1,4} + by_{1,4} + c \\ & + 0,5y_{2,4}^T A_{(2,b)} y_{2,4} + by_{2,4} + c + 0,5y_{3,4}^T A_{(3,b)} y_{3,4} + by_{3,4} + c \end{aligned} \right] \\ & + \left[\begin{aligned} & 0,5y_{0,5}^T A_{(0)} y_{0,5} + by_{0,5} + c + 0,5y_{1,5}^T A_{(1,b)} y_{1,5} + by_{1,5} + c \\ & + 0,5y_{2,5}^T A_{(2,a)} y_{2,5} + by_{2,5} + c + 0,5y_{3,5}^T A_{(3,a)} y_{3,5} + by_{3,5} + c \end{aligned} \right] \\ & + \left[\begin{aligned} & 0,5y_{0,6}^T A_{(0)} y_{0,6} + by_{0,6} + c + 0,5y_{1,6}^T A_{(1,b)} y_{1,6} + by_{1,6} + c \\ & + 0,5y_{2,6}^T A_{(2,a)} y_{2,6} + by_{2,6} + c + 0,5y_{3,6}^T A_{(3,b)} y_{3,6} + by_{3,6} + c \end{aligned} \right] \\ & + \left[\begin{aligned} & 0,5y_{0,7}^T A_{(0)} y_{0,7} + by_{0,7} + c + 0,5y_{1,7}^T A_{(1,b)} y_{1,7} + by_{1,7} + c \\ & + 0,5y_{2,7}^T A_{(2,b)} y_{2,7} + by_{2,7} + c + 0,5y_{3,7}^T A_{(3,a)} y_{3,7} + by_{3,7} + c \end{aligned} \right] \\ & + \left[\begin{aligned} & 0,5y_{0,8}^T A_{(0)} y_{0,8} + by_{0,8} + c + 0,5y_{1,8}^T A_{(1,b)} y_{1,8} + by_{1,8} + c \\ & + 0,5y_{2,8}^T A_{(2,b)} y_{2,8} + by_{2,8} + c + 0,5y_{3,8}^T A_{(3,b)} y_{3,8} + by_{3,8} + c \end{aligned} \right] \end{aligned} \right\} \quad (5.8) \\ s.a. \quad & \begin{aligned} & y_{0,1} = y_{0,2}, \quad y_{0,2} = y_{0,3}, \quad y_{0,3} = y_{0,4}, \quad y_{0,4} = y_{0,5}, \\ & y_{0,5} = y_{0,6}, \quad y_{0,6} = y_{0,7}, \quad y_{0,7} = y_{0,8}, \\ & y_{1,1} = y_{1,2}, \quad y_{1,2} = y_{1,3}, \quad y_{1,3} = y_{1,4}, \\ & y_{1,5} = y_{1,6}, \quad y_{1,6} = y_{1,7}, \quad y_{1,7} = y_{1,8}, \\ & y_{2,1} = y_{2,2}, \quad y_{2,3} = y_{2,4}, \quad y_{2,5} = y_{2,6}, \quad y_{2,7} = y_{2,8}, \\ & 0 \leq y_{t,e} \leq 10, \quad \forall t = 0, \dots, 3; \quad y_e = 0, \dots, 8; \end{aligned} \end{aligned}$$

donde los $y_{t,e} \in \mathbb{R}^2$ representan las variables de decisión correspondiente al período t y al escenario e . Por ejemplo, la variable $y_{1,2}$ corresponde al período 1 y al escenario 2, que agrupa a los nodos que pasan por las matrices $A_{(0)} \rightarrow A_{(1,a)} \rightarrow A_{(2,a)} \rightarrow A_{(3,b)}$. Esta trayectoria viene señalada en rojo en la figura 5.1.

Notemos que a diferencia del problema (5.7) en el cuál teníamos 4 variables $x_t \in \mathbb{R}^2$ cuando $t = 0, 1, 2, 3$, en el problema estocástico (5.8) existen 8 posibles escenarios y 32 variables de decisión $y_{t,e} \in \mathbb{R}^2$ cuando $t = 0, 1, 2, 3$ y $e = 1, \dots, 8$, razón por la cuál la función objetivo del problema estocástico viene multiplicado por la probabilidad $1/8$ correspondiente a cada escenario. Si bien este problema también puede ser resuelto usando técnicas clásicas de optimización no lineal, cuando la cantidad de eventos aleatorios crece, este enfoque ya no es sostenible. Por ejemplo, si quisieramos resolver un problema con $T = 99$ períodos y si consideramos una variación en la matriz A de 10 posibles eventos aleatorios, entonces el modelo estocástico implicaría resolver un problema de minimización que involucraría 10^{102} variables de decisión. Esto es justamente el *problema de la dimensionalidad* que comentamos en el capítulo introductorio. Para poder abordar esta dificultad, usaremos los métodos estudiados en el capítulo anterior; en particular, el algoritmo ADMM en forma de consenso correspondiente a la sección 3.5.

5.2.1. Solución numérica

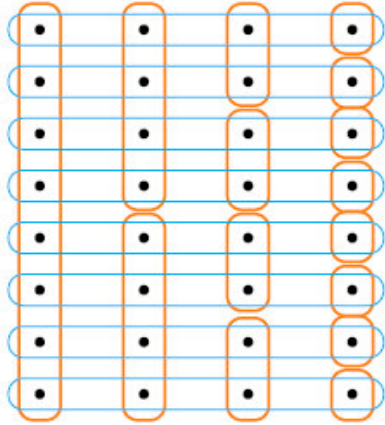


Figura 5.2: Problema de control estocástico en forma de consenso. Cada fila corresponde a una secuencia de resultados. Las casillas celestes muestran la estructura separable del objetivo y las casillas naranjas muestran las restricciones de consenso impuestas por el requisito de causalidad.

Para expresar este problema en forma de consenso debemos poner en correspondencia la secuencia de variables de decisión x_0, x_1, \dots, x_T con cada uno de las $|\Omega|^T$ secuencias de resultados. Luego imponemos la restricción de causalidad exigiendo que las decisiones en el período t con los mismos resultados hasta el tiempo t sean iguales. Esto se muestra en la figura 5.2, donde cada fila es una secuencia de resultados en particular, por lo que el objetivo (en azul) puede separarse en las filas de variables. Las restricciones de causalidad (en naranja) son restricciones de consenso, es decir, las variables dentro de cada grupo deben ser iguales. Ignorar las restricciones de causalidad nos da la solución bajo el supuesto de información perfecta sobre el futuro como vimos en la sección 4.3; esta, sería la mejor secuencia de decisiones si se conociera de antemano la secuencia de resultados completa.

La Figura 5.3 muestra el valor objetivo alcanzado por el ADMM respecto a cada iteración z^k (que satisface las restricciones de causalidad) progresando al valor óptimo. La línea discontinua inferior en color morado muestra el valor objetivo (obviamente menor) alcanzado por la solución bajo información perfecta y la línea discontinua superior en color verde muestra el valor objetivo alcanzado en valor esperado. Si bien el ADMM requiere alrededor de 6 iteraciones para alcanzar el

valor mínimo, podemos ver que estamos muy cerca de resolver el problema dentro de 4 iteraciones. La tabla 5.1 muestra un resumen de los tiempos de ejecución para obtener los resultados mostrados en la figura 5.3. En esta tabla se puede observar que el algoritmo ADMM es aproximadamente 27 veces más veloz que el algoritmo generado por el paquete CVX, más aún, es posible reducir mucho más el tiempo de ejecución del ADMM trabajando con procesadores en paralelo.

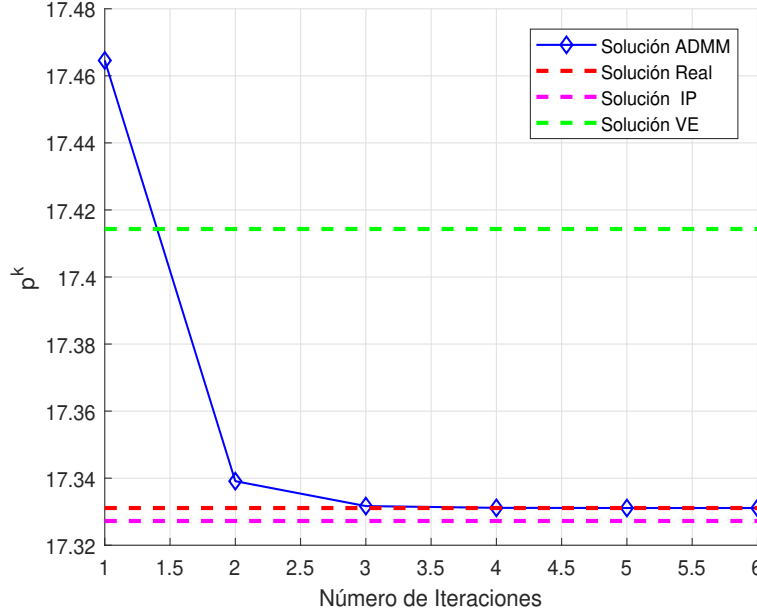


Figura 5.3: En azul, se muestra el valor objetivo alcanzado por la iteración z^k del ADMM (que satisface las restricciones de causalidad). La línea roja discontinua que se encuentre en medio de las dos líneas discontinuas es el valor óptimo del problema estocástico; la línea inferior morada es el valor óptimo alcanzado por la solución bajo información perfecta y la línea en verde representa la solución en valor esperado.

	ADMM	CVX		
		Sol. con restricciones de causalidad	Sol. bajo información perfecta	Sol. en valor esperado
tiempo(s)	0.064336	1.772479	1.303076	1.202051

Tabla 5.1: Tiempos de ejecución para cada tipo de solución.

5.3. Un segundo modelo para la gestión de la producción de E.E.

En esta sección modelizaremos la producción de energía eléctrica considerando N_{hid} centrales hidroeléctricas, N_{ter} centrales termoeléctricas y N_{des} de un tipo de central arbitraria, donde $N_{hid}, N_{ter}, N_{des} \in \mathbb{Z}^+$.

5.3.1. Hipótesis

Para simplificar el modelo, consideraremos el caso en el cuál se conoce la demanda de los clientes, el precio de mercado y además las centrales funcionan sin interrupciones causadas por defectos técnicos. También, ignoraremos los eventuales cierres de las centrales térmicas y trabajaremos para solo un posible escenario. Bajo estas consideraciones, este problema se simplifica enormemente con respecto a la realidad. En consecuencia, nuestra ecuación (4.36) tomará la forma

$$\begin{aligned}
 \text{Minimizar} \quad & \sum_{i,t} f_{u_i^{ter},t}(p_{u_i^{ter},t}) + \\
 & \sum_j C_{u_j^{hid}}(p_{u_j^{hid},1}, \dots, p_{u_j^{hid},T}) + \\
 & \sum_{k,t} f_{u_k^{des},t}(p_{u_k^{des},t}) \\
 \text{s.a.} \quad & \sum_i p_{u_i^{ter},t} + \sum_j p_{u_j^{hid},t} + \sum_k p_{u_k^{des},t} = D_t, \quad \forall t = 1, \dots, T, \\
 & \underline{p}_{u_i^{ter}} \leq p_{u_i^{ter},t} \leq \bar{p}_{u_i^{ter}}, \quad \forall t = 1, \dots, T, \\
 & \underline{p}_{u_j^{hid}} \leq p_{u_j^{hid},t} \leq \bar{p}_{u_j^{hid}}, \quad \forall t = 1, \dots, T, \\
 & \underline{p}_{u_k^{des}} \leq p_{u_k^{des},t} \leq \bar{p}_{u_k^{des}}, \quad \forall t = 1, \dots, T.
 \end{aligned} \tag{5.9}$$

donde:

- $i = 1, \dots, N_{ter}$;
- $j = 1, \dots, N_{hid}$;
- $k = 1, \dots, N_{des}$.

5.3.2. Construcción del algoritmo

Los algoritmos que usaremos aquí son los estudiados en los capítulos anteriores, los algoritmos de división de Douglas-Rachford (ADMM) y Forward-Backward (gradiente proximal), los cuáles son el corazón de nuestro proyecto.

La idea es descomponer el problema general (5.9) en subproblemas para cada central y aplicar el protocolo de consenso estudiado en 3.5.1. Consideremos entonces el problema:

$$\begin{aligned}
 \text{Minimizar} \quad & \sum_{\ell=1}^N f_{\ell}(x_{\ell}) \\
 \text{s.a.} \quad & \sum_{\ell=1}^N x_{\ell} = D, \\
 & L \leq x_{\ell} \leq U,
 \end{aligned} \tag{5.10}$$

donde $N = N_{ter} + N_{hid} + N_{des}$ representa a la cantidad total de centrales del cuál disponemos, $x_{\ell} \in \mathbb{R}^T$ con $\ell = 1, \dots, N$ por lo tanto:

- $x_i = (p_{u_i^{ter},1}, \dots, p_{u_i^{ter},T})$ cuando $i = 1, \dots, N_{ter}$.
- $x_{N_{ter}+j} = (p_{u_j^{hid},1}, \dots, p_{u_j^{hid},T})$ cuando $j = 1, \dots, N_{hid}$.
- $x_{N_{ter}+N_{hid}+k} = (p_{u_k^{des},1}, \dots, p_{u_k^{des},T})$ cuando $k = 1, \dots, N_{des}$.

y T es la cantidad de períodos del PPMT. Además, el problema de optimización restringida 5.10 podemos expresarlo en forma irrestricto del siguiente modo:

$$\text{Minimizar} \quad \sum_{\ell=1}^N f_{\ell}(x_{\ell}) + \iota_{\mathcal{C}}(x_1, \dots, x_N), \quad (5.11)$$

donde ι es la función indicatriz y $\mathcal{C} = \{(x_1, \dots, x_N) \mid L \leq x_{\ell} \leq U, x_1 + \dots + x_N = D\}$.

La separabilidad de f y $g = \iota$, y las particiones \mathcal{P} y \mathcal{Q} son los mismos que en (3.2). De aquí, el algoritmo resultante será:

Algoritmo de Consenso Proximal

- ▶ $x_i^{k+1} := \text{prox}_{\alpha f_i}(z_i^k - u_i^k)$
- ▶ $z_j^{k+1} := \Pi_{\mathcal{C}}(x_j^{k+1} + u_j^k)$
- ▶ $u_i^{k+1} := u_i^k + x_i^{k+1} - z_i^{k+1}.$

Este algoritmo requiere el cálculo de N operadores proximales, los cuáles pueden tener una representación trivial cuando por ejemplo son aproximados como casos particulares del operador proximal de una función cuadrática, es decir, dado una función cuadrática $f(x) = (0,5)x^T A x + b^T x + c$, donde $A \in S_+^n$ entonces su operador proximal es de la forma $\text{prox}_{\alpha f}(v) = (I + \alpha A)^{-1}(v - \alpha b)$.

Por otro lado, la proyección sobre \mathcal{C} involucra n proyecciones independientes sobre la intersección del Hiperplano $H_{m,d} = \{x \in \mathbb{R}^N \mid m^T x = d\}$ y el Box $B[L, U] = \{x \in \mathbb{R}^N \mid L \leq x \leq U\}$. Esta proyección puede ser calculado usando el método de la bisección.

5.3.3. Solución numérica

Para nuestra experimentación numérica consideraremos lo siguiente:

- 5 centrales termoelectricas ($N_{ter} = 5$) con funciones de costo lineal, es decir, $f_{u_i^{ter},t}(p_{u_i^{ter},t}) = c_{u_i^{ter}} p_{u_i^{ter},t}$, donde $c_{u_i^{ter}} > 0$ para todo $i = 1, \dots, N_{ter}$.
- 3 centrales hidroelectricas ($N_{hid} = 3$) con funciones de costo constante como se sugiere en la sección 5.1.2, es decir, $C_{u_j^{hid}}(p_{u_j^{hid},1}, \dots, p_{u_j^{hid},T}) = T * c_{u_j^{hid}}$, donde $c_{u_j^{hid}} > 0$ para todo $j = 1, \dots, N_{hid}$ y T es la cantidad de períodos.
- 4 centrales arbitrarias ($N_{des} = 4$) con funciones de costo cuadrático, es decir, $f_{u_k^{des},t}(p_{u_k^{des},t}) = 0,5 p_{u_k^{des},t}^2$, para todo $k = 1, \dots, N_{des}$.

Por lo tanto, el problema a minimizar sería el siguiente:

$$\begin{aligned}
& \text{Minimizar} \quad \sum_{i,t} c_{u_i^{ter}} p_{u_i^{ter},t} + \sum_{j,t} c_{u_j^{hid}} + \sum_{k,t} 0,5 p_{u_k^{des},t}^2 \\
& \text{s.a.} \quad \sum_i p_{u_i^{ter},t} + \sum_j p_{u_j^{hid},t} + \sum_k p_{u_k^{des},t} = D_t, \quad \forall t = 1, \dots, T, \\
& \quad \underline{P}_{u_i^{ter}} \leq p_{u_i^{ter},t} \leq \bar{P}_{u_i^{ter}}, \quad \forall t = 1, \dots, T, \\
& \quad \underline{P}_{u_j^{hid}} \leq p_{u_j^{hid},t} \leq \bar{P}_{u_j^{hid}}, \quad \forall t = 1, \dots, T, \\
& \quad \underline{P}_{u_k^{des}} \leq p_{u_k^{des},t} \leq \bar{P}_{u_k^{des}}, \quad \forall t = 1, \dots, T.
\end{aligned} \tag{5.12}$$

- $i = 1, \dots, N_{ter}$;
- $j = 1, \dots, N_{hid}$;
- $k = 1, \dots, N_{des}$.

Adicionalmente, tomando como referencia las fichas técnicas de las diferentes tecnologías para la producción de E.E. publicadas en [Osinermin, 2016, 2018], podemos establecer como parámetros para la ecuación 5.12 a los datos que figuran en las tablas 5.5, 5.2, 5.3 y 5.4 que mostramos a continuación:

	Datos para las Centrales Termoeléctricas (CT)		
	$\underline{P}_{u_i^{ter}}$ (MW/30min.)	$\bar{P}_{u_i^{ter}}$ (MW/30min.)	$c_{u_i^{ter}}$ (unid. monetarias/MW)
CT-1	1.3842	2116.6	105.5682
CT-2	0.0549	1739.8	132.3096
CT-3	7.3852	2291.1	193.3615
CT-4	8.1079	2205.6	185.2758
CT-5	5.6472	2237.9	168.6773

Tabla 5.2: *Fuente propia.*

	Datos para las Centrales Hidroeléctricas (CH)		
	$\underline{P}_{u_j^{hid}}$ (MW/30min.)	$\bar{P}_{u_j^{hid}}$ (MW/30min.)	$c_{u_j^{hid}}$ (unid. monetarias)
CH-1	668.3011	1667.2	2736.5
CH-2	637.7125	2179.7	1122.8
CH-3	699.5203	1579.6	2769.6

Tabla 5.3: *Fuente propia.*

	Datos para las Centrales arbitrarias (CD)	
	$\underline{P}_{u_k}^{des}$ (MW/30min.)	$\overline{P}_{u_k}^{des}$ (MW/30min.)
CD-1	8.2593	1934.1
CD-2	9.1592	1020.8
CD-3	2.6735	2187.1
CD-4	2.9602	1603.3

Tabla 5.4: *Fuente propia.*

En consecuencia, nuestro problema a minimizar sería el siguiente:

$$\begin{aligned}
\text{Minimizar} \quad & \sum_{t=1}^T \left[105,5682 p_{u_1^{ter},t} + 132,3096 p_{u_2^{ter},t} + 193,3615 p_{u_3^{ter},t} + 185,2758 p_{u_4^{ter},t} + 168,6773 p_{u_5^{ter},t} \right] + \\
& \sum_{t=1}^T [2736,5 + 1122,8 + 2769,6] + \sum_{t=1}^T 0,5 \left[p_{u_1^{des},t}^2 + p_{u_2^{des},t}^2 + p_{u_3^{des},t}^2 + p_{u_4^{des},t}^2 \right] \\
\text{s.a.} \quad & \sum_i p_{u_i^{ter},t} + \sum_j p_{u_j^{hid},t} + \sum_k p_{u_k^{des},t} = D_t, \quad \forall t = 1, \dots, T, \\
& 1,3842 \leq p_{u_1^{ter},t} \leq 2116,6, \quad \forall t = 1, \dots, T, \\
& 0,0549 \leq p_{u_2^{ter},t} \leq 1739,8, \quad \forall t = 1, \dots, T, \\
& 7,3852 \leq p_{u_3^{ter},t} \leq 2291,1, \quad \forall t = 1, \dots, T, \\
& 8,1079 \leq p_{u_4^{ter},t} \leq 2205,6, \quad \forall t = 1, \dots, T, \\
& 5,6472 \leq p_{u_5^{ter},t} \leq 2237,9, \quad \forall t = 1, \dots, T, \\
& 668,3011 \leq p_{u_1^{hid},t} \leq 1667,2, \quad \forall t = 1, \dots, T, \\
& 637,7125 \leq p_{u_2^{hid},t} \leq 2179,7, \quad \forall t = 1, \dots, T, \\
& 699,5203 \leq p_{u_3^{hid},t} \leq 1579,6, \quad \forall t = 1, \dots, T, \\
& 8,2593 \leq p_{u_1^{des},t} \leq 1934,1, \quad \forall t = 1, \dots, T, \\
& 9,1592 \leq p_{u_2^{des},t} \leq 1020,8, \quad \forall t = 1, \dots, T, \\
& 2,6735 \leq p_{u_3^{des},t} \leq 2187,1, \quad \forall t = 1, \dots, T, \\
& 2,9602 \leq p_{u_4^{des},t} \leq 1603,3, \quad \forall t = 1, \dots, T.
\end{aligned} \tag{5.13}$$

Para los datos correspondientes a D_t , cuando $t = 1, \dots, T$; usaremos los datos reales de la demanda energética del mes de enero y febrero del año 2019 (con una frecuencia de 30 min.), los cuáles fueron decargados de la página oficial del COES y corresponden a la tercera columna de la tabla 5.5 (PROG. DIARIA (MW.)). Esta serie temporal podemos graficarla usando el *software R con el package dygraphs*, el cuál es mostrada en la figura 5.4. Se puede observar en dicha figura que la demanda energética (PROG. DIARIA), muchas veces suele ser mayor que la cantidad despachada (EJECUTADA). Para observar esto, hacemos un zoom de la figura 5.4, el cuál es mostrada en la figura 5.5.

Después de ejecutar el software CVX y los algoritmos de división para $T = 240$ (los 5 primeros días del mes de enero de 2019) resulta que el software CVX empleó 1.558773 segundos para alcanzar el valor óptimo, mientras que los algoritmos de división: Douglas-Rachford (ADMM) empleó

Datos de la demanda y despacho		
Fecha	EJECUTADO (MW)	PROG. DIARIA (MW)
01/01/2019 00:30	5564.8221	5685.3023
01/01/2019 01:00	5485.4548	5553.2813
01/01/2019 01:30	5402.9336	5438.3295
01/01/2019 02:00	5253.6771	5360.6938
⋮	⋮	⋮
11/02/2019 00:00	5428.2784	5688.9315

Tabla 5.5: Datos correspondientes al mes de enero y los 10 primeros días del mes de febrero (freq = 30min.).

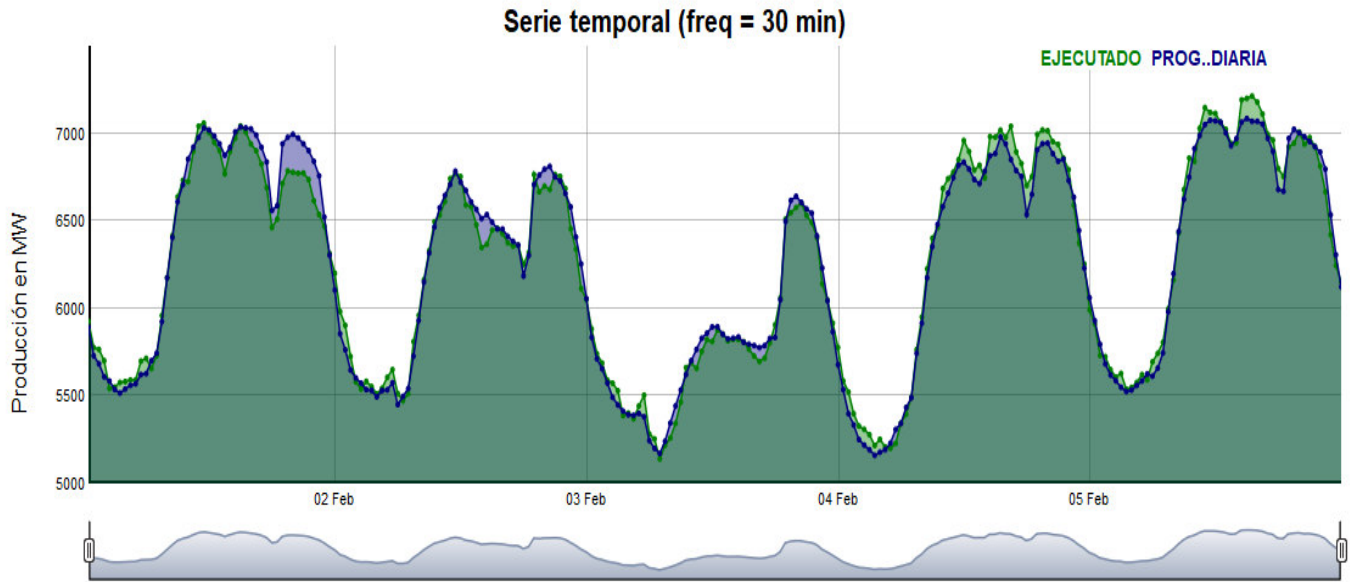


Figura 5.4: Demanda y despacho de la potencia eléctrica de los 5 primeros días del mes de febrero del 2019, con una frecuencia de 30 min.

44 iteraciones con un tiempo de 1.168757 segundos y los algoritmos Forward-Backward (gradiente proximal y su versión acelerada) emplearon 69 iteraciones en 1.453028 segundos y 62 iteraciones en 1.304973 segundos respectivamente. Debemos señalar que los algoritmos de división empleados, si bien están paralelizados, han sido ejecutados en un solo cpu dejando así la posibilidad de poder reducir mucho más el tiempo de ejecución. Los resultados de la convergencia pueden observarse en la figura 5.6.

Por otro lado, el valor óptimo obtenido fué de $1,4962 \times 10^7$ unidades monetarias para estos 5 días. La tabla 5.6 muestra la cantidad de potencia que debe producir cada central para los primeros tres períodos del ciclo. Como era de esperar, las centrales hidroeléctricas son las que trabajan casi al 100 % de su capacidad, esto es debido a que su costo de producción es muy baja respecto a las otras tecnologías. Enfatizamos que estos primeros resultados son aproximaciones de los casos reales que tienen como objetivo ilustrar las aplicaciones de los algoritmos estudiados. Obviamente, existen restricciones adicionales que evitan que las centrales hidroeléctricas trabajen al 100 % de su capacidad. Una implementación tomando en cuenta todas las restricciones del sistema escapa a

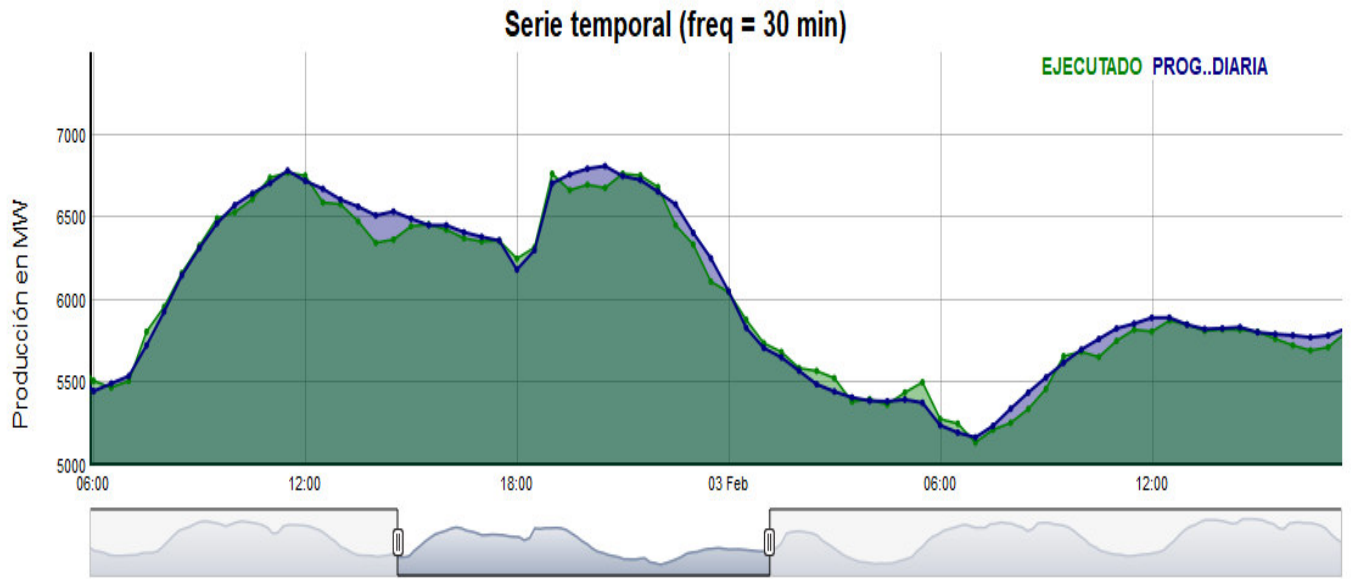


Figura 5.5: *Demanda y despacho de la potencia eléctrica entre las 6 a.m del 2 de febrero y las 6 p.m. del 3 de febrero del 2019.*

los objetivos de este trabajo por dos razones fundamentales: la primera, esto implicaría conocer en profundidad los detalles técnicos del funcionamiento de cada tecnología y la segunda, que es la más importante, dicha implementación requiere de procesadores con alta performance para procesar la cantidad de variables que involucraría el modelamiento real considerando las incertidumbres. A manera de ejemplo, si tuviésemos que procesar el modelo con 10 escenarios y $T = 365$ (donde cada período representa un día), entonces tendríamos como mínimo 10^{365} variables. No obstante, actualmente ya existe la tecnología para procesar esta cantidad de información.

Concluimos esta experimentación ejecutando los algoritmos de división para los 31 días del mes de enero ($T = 1488$) cambiando la escala de las unidades monetarias (multiplicandolas por 10). El valor óptimo hallado fué de $2,6083 \times 10^8$. Esta vez el software CVX empleó 3.374650 segundos, el algoritmo ADMM empleó 13 iteraciones en un tiempo de 1.988358 segundos, mientras que los algoritmos del gradiente proximal y su versión acelerada emplearon 18 iteraciones en 2.677171 segundos y 16 iteraciones en 2.271945 segundos respectivamente.

	Períodos (01/01/2019)		
	00 : 30 – 01 : 00	01 : 00 – 01 : 30	01 : 30 – 02 : 00
CT-1	1.4000	1.4000	1.4000
CT-2	0.1000	0.1000	0.1000
CT-3	7.4000	7.4000	7.4000
CT-4	8.1000	8.1000	8.1000
CT-5	5.6000	5.6000	5.6000
CH-1	1667.2	1667.2	1656.0
CH-2	2179.7	2179.7	2168.3
CH-3	1579.6	1579.6	1568.4
CD-1	59.100	26.100	8.3000
CD-2	59.100	26.100	9.2000
CD-3	59.100	26.100	2.7000
CD-4	59.100	26.100	3.0000

Tabla 5.6: *Producción para los tres primeros períodos por cada central de generación eléctrica (MW/30 min.).*

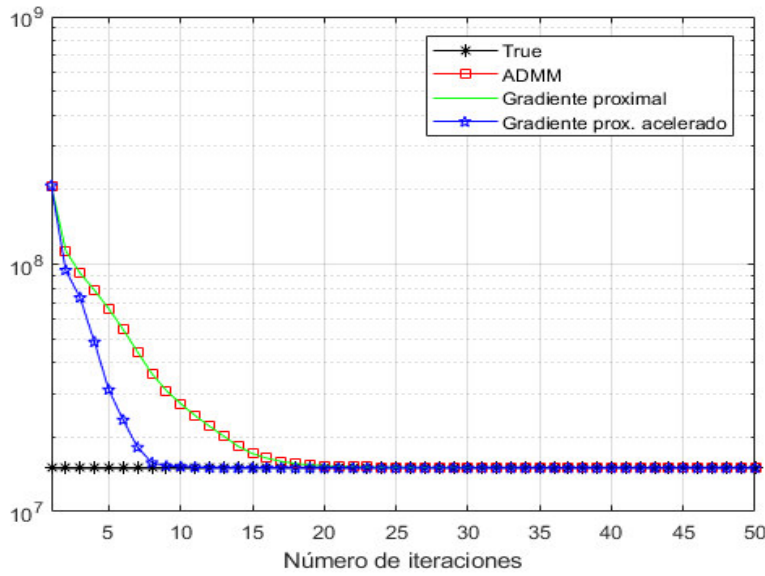


Figura 5.6: *Velocidad de convergencia de los tres algoritmos de división.*

5.4. Conclusiones

Sobre la base de los objetivos generales y específicos planteados en esta tesis, en el Capítulo 4 hemos resaltado los tres acoplamientos naturales del problema de gestión de la producción en un entorno incierto, brindando además varias visiones del sistema global, cada una de ellas basada en una definición diferente del subsistema. Esta enumeración sistemática de los acoplamientos, nos ha permitido contemplar nuevas formas de descomponer el problema, los cuáles se han resumido hasta ahora en el desacoplamiento espacial, temporal y aleatorio. Hemos mostrado también la estructura del árbol de escenarios como un método de discretización para modelizar las incertidumbres en la adquisición de los datos, enfatizando algunas de sus debilidades y dando paso de este modo al pro-

ceso de discretización propuesta por Nadaraya y Watson desarrollado en [Lenoir, 2008]. El efecto de la dicretización y las condiciones físicas del sistema, trajo como consecuencia dos problemas a enfrentar: por un lado, la aparición de funciones no diferenciables y por el otro, el problema de la dimensionalidad de las variables subyacentes.

La primera problemática es subsanado usando los instrumentos estudiados en el capítulo 2, en el cuál, introducimos la teoría de los operadores monótonos maximales mostrando su fuerte conexión con el análisis convexo. Esta conexión, nos permitió afrontar el problema de la no diferenciable por medio de un proceso de regularización, es decir; dado una función convexa no diferenciable, mostramos que era posible construir una función diferenciable llamada la envolvente o parametrización de Moreau-Yosida, el cuál, tiene los mismos mínimos que la función original. Este capítulo fué también el punto de partida para el estudio de las propiedades de convergencia de diferentes algoritmos que vimos posteriormente.

El problema de la dimensionalidad la afrontamos en el capítulo 3 dentro del campo de los “grandes sistemas”. Aquí, presentamos la idea del método de *descomposición-coordinación*. Este paradigma nos permitió introducir diferentes familias de algoritmos resultante de los métodos *splitting*: los algoritmos de Forward-Backward, Peaceman-Rachford, Douglas-Rachford splitting, y los derivados de estos tales como: el método del gradiente, el gradiente proximal y su versión acelerada y el más importante, el ADMM. Al ser todas instancias de la iteración del punto fijo, concluimos que estas convergen al menos linealmente a una solución siempre que esta exista. Además, vimos que existen formas de como acelerar el proceso de convergencia en casos específicos. No obstante, los procesos de acelaración dependerán de las características de las funciones implicadas en el problema. Finalmente, en el capítulo 5 sintetizamos todos los conceptos antes estudiados mediante la implementación de dos modelos sencillos (junto con su solución numérica) para la gestión de la producción de la energía eléctrica. El primer modelo tuvo como objetivo mostrar como introducir los efectos aleatorios en el análisis, mientras que en el segundo nos focalizamos más en las restricciones de la demanda energética y el PPMT.

Concluimos esta tesis mencionando que existen trabajos como las de [Lenoir, 2008], [Themelis & Patrinos, 2016], entre otros, en donde demuestran que bajo ciertas condiciones, la convergencia de los algoritmos splitting es superlineal, pero aún no se ha alcanzado la convergencia cuadrática. Una alternativa aún fértil para afrontar este problema y muchos otros, se presenta en la técnica *splitting* basada en las distancias de Bregman. Este marco está motivado por la necesidad de resolver problemas en los espacios de Banach, en donde las nociones estándar de operador resolvente y proximal ya no son apropiadas, así como por consideraciones numéricas en los espacios euclidianos, ya que algunos operadores proximales pueden ser más fáciles de implementar en la forma Bregman o incluso, algunas de las funciones pueden tener propiedades más explotables cuando se examinan a través de las distancias de Bregman [Bauschke *et al.*, 2016; Combettes & Van Nguyen, 2015; Van Nguyen, 2017]. Como última palabra, vale la pena mencionar que aún sigue siendo un gran desafío construir teorías equivalentes sólidas que permitan derivar algoritmos estables en el vasto mundo de los problemas no convexos y no monótonos.

Apéndice

Capítulo A

Revisión del análisis convexo

El problema matemático subyacente que se aborda en esta tesis se sitúa en el área de la optimización matemática, cuyo objetivo fundamental en líneas generales, es la búsqueda de la *mejor decisión* de entre muchas alternativas. Matemáticamente, esto puede reducirse al estudio del problema del mínimo, es decir:

Dado un conjunto arbitrario $\mathbb{X} \neq \emptyset$ y una función $f : \mathbb{X} \rightarrow \overline{\mathbb{R}} = \mathbb{R} \cup \{\pm\infty\}$, el objetivo es encontrar el $\min\{f(x) : x \in \mathbb{X}\}$, donde $\operatorname{argmin}\{f(x), x \in \mathbb{X}\}$ representa el punto donde f toma dicho valor mínimo.

Vemos claramente que en este problema la finalidad es “encontrar el mínimo valor de $f(x)$ cuando x varía en el conjunto \mathbb{X} ”. Entender bien esta última frase encerrada entre comillas es trascendental para responder preguntas del tipo: ¿Existe un $x \in \mathbb{X}$ tal que $\min\{f(x) : x \in \mathbb{X}\}$ también exista?.. y si la respuesta es positiva entonces ¿cómo encontrar el valor de x ?.. y si la respuesta es negativa ¿sería posible reemplazar $\min\{f(x) : x \in \mathbb{X}\}$ por $\inf\{f(x) : x \in \mathbb{X}\}$?

La respuesta a estas interrogantes sin duda radica en la estructura del conjunto \mathbb{X} , es decir, dado

$$f : \mathbb{X} \rightarrow \overline{\mathbb{R}}$$

donde el conjunto de partida puede ser:

- $\mathbb{X} \rightarrow$ conjunto
- \rightarrow espacio con una Noción de convergencia
- \rightarrow espacio Topológico
- \rightarrow espacio Métrico
- \rightarrow espacio de Banach
- \rightarrow espacio de Hilbert
- $\rightarrow \mathbb{R}^n$

La situación más insignificante tendría lugar cuando \mathbb{X} es un conjunto arbitrario sin ninguna estructura en particular, en cuyo caso en realidad no hay mucho que hacer. Por otro lado, la situación más completa tendría lugar cuando $\mathbb{X} = \mathbb{R}^n$, en donde se aplican todas las reglas del análisis 1 y 2 y cuyo estudio termina dentro de ese contexto. Ubiquémonos a continuación en el caso abstracto cuando \mathbb{X} es un *espacio con una noción de convergencia*.

Definición A.1. Dado un conjunto \mathbb{X} diferente del vacío (de aquí en adelante esto quedará implícito), una **noción de convergencia** en \mathbb{X} es cualquier subconjunto de $Suc(\mathbb{X}) \times \mathbb{X}$, donde

$$Suc(\mathbb{X}) = \{\mathcal{S} : \mathbb{N} \rightarrow \mathbb{X}\}$$

Una observación a tener en cuenta en esta definición es que tal subconjunto puede ser cualquiera, por ejemplo, sucesiones constantes que convergen a otro valor, sucesiones que tienen infinitos límites, etc. Veremos que todo lo que mencionamos a continuación funcionan perfectamente en este contexto.

Definición A.2. Un subconjunto $\mathbf{K} \subseteq \mathbb{X}$ se dice que es compacto (por sucesiones) si cada sucesión definida en \mathbf{K} admite una subsucesión que converge a un elemento en \mathbf{K} .

Definición A.3. Sea $f : \mathbb{X} \rightarrow \overline{\mathbb{R}}$ una función, se dice

- **Continua**, si para cada sucesión: $x_n \rightarrow x_\infty$ en \mathbb{X} vale

$$f(x_n) \rightarrow f(x_\infty).$$

- **Semicontinua inferiormente (s.c.i.)**, si para cada sucesión: $x_n \rightarrow x_\infty$ en \mathbb{X} vale

$$\liminf_{n \rightarrow \infty} f(x_n) \geq f(x_\infty).$$

Estas definiciones conducen automáticamente al teorema de Weierstrass.

Teorema A.1 (Teorema de Weierstrass). Sea $f : \mathbb{X} \rightarrow \overline{\mathbb{R}}$ una función, supongamos que

(i) \mathbb{X} es compacto.

(ii) f es s.c.i.,

ambos, (i) y (ii) respecto a la misma noción de convergencia; entonces existe

$$\min\{f(x) : x \in \mathbb{X}\}.$$

Demostración. Denotemos por $I := \inf\{f(x) : x \in \mathbb{X}\}$, este existirá necesariamente en $\mathbb{R} \cup \{-\infty\}$. Del análisis en \mathbb{R} sabemos que existe $\{y_n\} \subseteq f(\mathbb{X})$ tal que

$$y_n \rightarrow I.$$

Por la definición de imagen, existe $\{x_n \subseteq \mathbb{X}\}$ tal que $y_n = f(x_n)$. Por otro lado, dado que \mathbb{X} es compacto, existe $x_{n_k} \rightarrow x_\infty$. Entonces,

$$I \leq f(x_\infty) \leq \liminf_{k \rightarrow \infty} f(x_{n_k}) = \liminf_{k \rightarrow \infty} y_{n_k} = \lim_{n \rightarrow \infty} y_n = I.$$

Por lo tanto $I = f(x_\infty)$, y de aquí tenemos que $I \in \mathbb{R}$. □

Existe una versión generalizada del teorema de Weierstrass, el cuál se desprende de la siguiente definición.

Definición A.4. Se dice que una función $f : \mathbb{X} \rightarrow \overline{\mathbb{R}}$ es **coerciva** si

$$\exists \mathbf{K} \subseteq \mathbb{X} \text{ compacto} / \inf\{f(x) : x \in \mathbf{K}\} = \inf\{f(x) : x \in \mathbb{X}\}.$$

Corolario A.1. Si $f : \mathbb{X} \rightarrow \overline{\mathbb{R}}$ es coerciva y s.c.i., entonces existe el mínimo.

Demostración. Se desprende inmediatamente de la condición

$$\inf_{x \in \mathbb{X}} f(x) = \inf_{x \in \mathbf{K}} f(x) = \min_{x \in \mathbf{K}} f(x).$$

□

Corolario A.2. Si $f : \mathbb{X} \rightarrow \overline{\mathbb{R}}$ es s.c.i. y existe un subnivel $\mathfrak{M} = \{x \in \mathbb{X} : f(x) \leq M\}$ contenido en un compacto y no vacío, es decir

$$\exists M \in \mathbb{R}, \exists \mathbf{K} \subseteq \mathbb{X} \text{ compacto} / \emptyset \neq \{x \in \mathbb{X} : f(x) \leq M\} \subseteq \mathbf{K}.$$

Entonces, existe el mínimo.

Demostración. Se desprende inmediatamente del corolario anterior aplicado a este compacto \mathbf{K} . □

Como observación general debemos mencionar que una respuesta al problema de encontrar el mínimo, se reduce entonces al teorema de Weierstrass, es decir; la astucia está en encontrar una noción de convergencia respecto al cuál se satisfagan las dos hipótesis del teorema de Weierstrass A.1, el cuál debemos mencionar que no es un asunto trivial ya que dichas hipótesis son antitéticas en el sentido que:

- si tengo tantas sucesiones convergentes, entonces (i) es fácil y (ii) es difícil,
- si tengo pocas sucesiones convergentes, entonces (ii) es fácil y (i) es difícil,

Pero, ¿qué sucede si ya no se cumple el teorema de Weierstrass?. Entonces, la respuesta al problema en cuestión dependerá de las condiciones particulares que definen dicho problema. Sin embargo, en esta tesis nos centraremos principalmente en el problema de encontrar el mínimo cuando $\mathbb{X} = \mathbb{R}^n$ es el espacio euclidiano, el cuál es un buen cuadro para la formulación de los problemas de optimización sobre el cuál estamos interesados y, que además, permite cubrir muchas aplicaciones de interés en diferentes campos de las ciencias e ingeniería donde a veces es innecesario el estudio en espacios de dimensión infinita. Por lo tanto, a continuación hacemos un breve repaso de los principales resultados del análisis convexo en \mathbb{R}^n . Todo el contenido que viene a continuación está basado en los trabajos desarrollados en [Barbu & Precupanu, 2012; Ekeland & Temam, 1999; Hiriart-Urruty & Lemaréchal, 2013], y así como en una de las obras “fundadoras”, la de R. Tyrrell Rockafellar [Rockafellar, 2015].

A.0.1. Conjuntos convexos

Sean \mathcal{X} un espacio vectorial real, \mathcal{C} y \mathcal{D} subconjuntos de \mathcal{X} , $z \in \mathcal{X}$ y $\lambda \in \mathbb{R}$. Tenemos los siguientes conjuntos especiales:

- $\mathcal{C} + \mathcal{D} = \{x + y / x \in \mathcal{C}, y \in \mathcal{D}\}.$
- $\mathcal{C} - \mathcal{D} = \{x - y / x \in \mathcal{C}, y \in \mathcal{D}\}.$
- $z + \mathcal{C} = \{z\} + \mathcal{C}.$
- $\mathcal{C} - z = \mathcal{C} - \{z\}.$

► $\lambda C = \{\lambda x / x \in C\}$.

Si además Λ es un subconjunto no vacío de \mathbb{R} , entonces

► $\Lambda C = \bigcup_{\lambda \in \Lambda} \lambda C$.

► $\Lambda z = \lambda\{z\} = \{\lambda z / \lambda \in \Lambda\}$.

Así también, si ahora tomamos $u \in X \setminus \{0\}$ decimos que C es un “*rayo*”, o una “*línea*” si

$$C = \mathbb{R}_+ u \quad \text{o} \quad C = \mathbb{R}u$$

respectivamente. Veamos ahora algunos conjuntos más elaborados

Definición A.5. Sea X un espacio vectorial real. Decimos que un subconjunto \mathcal{A} de X es un **subespacio afín** si

$$\mathcal{A} \neq \emptyset \quad \text{y} \quad \alpha x + (1 - \alpha)y \in \mathcal{A}$$

para todo $x, y \in \mathcal{A}$ y para todo $\alpha \in \mathbb{R}$

Una forma de entender esta definición es ver a un espacio afín como aquel que contiene a todas las rectas que pasan por dos cualesquiera de sus puntos x e y .

Por otro lado, como $\mathcal{A} \neq \emptyset$, del algebra lineal sabemos que la intersección de todos los subespacios vectoriales (llamado también “*lineales*”) conteniendo a \mathcal{A} , i.e., el subespacio vectorial más pequeño de X conteniendo a \mathcal{A} es llamado el **span** de \mathcal{A} el cuál es denotado por $\text{span}(\mathcal{A})$. De manera análoga, dado que la intersección de subespacios afines es nuevamente un subespacio afín, y además el conjunto de los subespacios afines que contienen a \mathcal{A} es no vacío (ya que X es uno de tales), entonces la siguiente definición tiene sentido.

Definición A.6. La **envolvente afín** de un subconjunto \mathcal{A} , denotado por $\varsigma(\mathcal{A})$, es el más pequeño subespacio afín conteniendo a \mathcal{A} .

No es difícil demostrar que

$$\varsigma(\mathcal{A}) = \left\{ x \mid n \in \mathbb{N}, x = \sum_{i=1}^n \alpha_i x_i, x_i \in \mathcal{A}, \sum_{i=1}^n \alpha_i = 1 \right\}.$$

Definición A.7. Sea X un espacio vectorial real. Un subconjunto \mathcal{C} de X es **convexo** si

$$\alpha x + (1 - \alpha)y \in \mathcal{C}$$

para todo $x, y \in \mathcal{C}$ y para todo $\alpha \in [0, 1]$.

Dicho de otro modo, un subconjunto convexo contiene siempre el segmento $[x, y]$ que une dos de sus puntos x e y .

Definición A.8. Se denomina **simplex** de \mathbb{R}^n al subconjunto

$$\mathbb{S}_n \stackrel{\text{def}}{=} \left\{ \alpha \in \mathbb{R}^n \mid \alpha_i \geq 0, i = 1, \dots, n, \sum_{i=1}^n \alpha_i = 1 \right\}.$$

La figura A.1 representa el **simplex** \mathbb{S}_3 de \mathbb{R}^3 .

Definición A.9. Llamamos **combinación convexa** de “ n ” puntos $\{x_i\}_{i=1,\dots,n}$ a todo punto x obtenido por la fórmula

$$x = \sum_{i=1}^n \alpha_i x_i$$

con $\alpha \in \mathbb{S}_n$.

Se puede demostrar rápidamente que una combinación convexa de n puntos puede ser calculada recursivamente por $n - 1$ combinaciones convexas de 2 puntos; lo cuál demuestra que en la definición A.7, podemos reemplazar de manera equivalente la combinación convexa de 2 puntos por n puntos arbitrarios. De hecho, en un espacio de dimensión finita n , es inútil considerar las combinaciones convexas de más de $n + 1$ puntos, como lo demuestra el teorema de Caratheodory el cuál se desprende inmediatamente del siguiente lema:

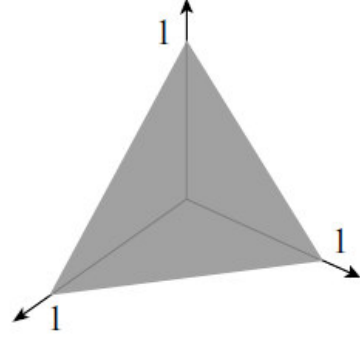


Figura A.1: El **simplex** de \mathbb{R}^3 .

Lema A.1 (Caratheodory). *En un espacio vectorial de dimensión finita n , todas las combinaciones convexas de m puntos con $m > n + 1$ se puede reducir a una combinación convexa de $n + 1$ puntos como máximo.*

Demostración. Ver [Cohen, 2000]. □

Definición A.10. Sea X un espacio vectorial real. La **envolvente convexa** de un subconjunto A de X , es el más pequeño conjunto convexo (en el sentido de la inclusión) que contiene a A , el cuál es denotado por $\mathcal{Conv}(A)$.

Notemos que la construcción de la envolvente convexa propuesta en la definición A.10 (intersección de convexas conteniendo a A) es una construcción “externa”. El teorema que viene a continuación propone una construcción “interna”.

Teorema A.2. *La envolvente convexa de A es igual a conjunto formado por las combinaciones convexas de los elementos del mismo A .*

Demostración. Ver [Cohen, 2000]. □

Ahora estamos listos para enunciar el teorema de Caratheodory.

Teorema A.3 (Caratheodory’s theorem). *En un espacio vectorial de dimensión n , la envolvente convexa de un subconjunto A es igual al conjunto de las combinaciones convexas de $n + 1$ puntos de A .*

Demostración. Este teorema es un simple corolario del lema A.1. □

Cuando \mathcal{X} es un espacio de Hilbert \mathbb{R}^n , existe una clase de conjuntos muy importante que jugarán un papel esencial en la formulación de restricciones de desigualdad, estos son los “conos”.

Decimos que C es un “cono” si;

$$C = \mathbb{R}_+ C, \quad (\text{A.1})$$

Un cono es, por lo tanto, una unión de semirectas cerradas desde el origen. La figura A.2 representa un cono en \mathbb{R}^3 (de aquellos no convexos).

Es fácil observar que \mathbb{R}^n es también un cono, además que los subespacios vectoriales son también conos y la intersección de una familia de conos es nuevamente un cono. Por lo tanto, tenemos la siguiente definición:

Definición A.11. Sea C un subconjunto de \mathbb{R}^n . La **envolvente cónica** de C es la intersección de todos los conos en \mathbb{R}^n que contienen a C , i.e., el más pequeño cono en \mathbb{R}^n conteniendo a C . Este viene denotado por $\text{cone}(C)$. De igual modo, la **envolvente cónica cerrada** de C es el más pequeño cono cerrado en \mathbb{R}^n conteniendo a C y viene denotado por $\overline{\text{cone}}(C)$.

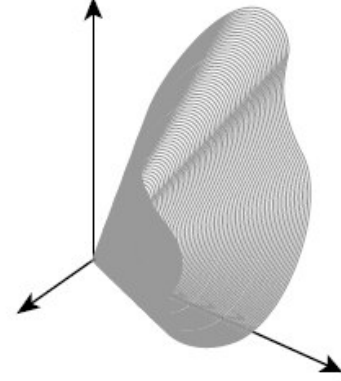


Figura A.2: Un cono no convexo de \mathbb{R}^3 .

Recordemos ahora que el interior de un subconjunto C de \mathbb{R}^n puede ser expresado como

$$\text{int}(C) = \{x \in C / (\exists \rho \in \mathbb{R}_{++}) \quad B(0; \rho) \subset C - x\}. \quad (\text{A.2})$$

A.0.2. Funciones convexas

En cierta medida, este tema está estrechamente relacionada con la sección anterior (conjuntos convexos) ya que en efecto, podemos asociar a una función convexa un subconjunto convexo de $\mathbb{R}^n \times \mathbb{R}$ llamado “epígrafo”, el cuál es convexo si y solamente si la función es convexa; inversamente, podemos asociar a un subconjunto de \mathbb{R}^n conformado por funciones a valores en la recta extendida que son convexas si (o si y solo si) dicho subconjunto es convexo.

Definición A.12. Sea $f : \mathbb{R}^n \rightarrow \overline{\mathbb{R}}$ una función, tenemos que:

- La **gráfica** de f está dada por:

$$\text{gra}(f) = \{(x, \xi) \in \mathbb{R}^n \times \mathbb{R} \mid f(x) = \xi\}.$$

- El **epígrafo** de f es el subconjunto:

$$\text{epi}(f) = \{(x, \xi) \in \mathbb{R}^n \times \mathbb{R} \mid f(x) \leq \xi\}.$$

- El **epígrafo estricto** de f es el subconjunto:

$$\text{epi}_s(f) = \{(x, \xi) \in \mathbb{R}^n \times \mathbb{R} \mid f(x) < \xi\}.$$

- El conjunto de nivel inferior de f de altura $\xi \in \mathbb{R}$ es:

$$\text{lev}_{\leq \xi} f = \{x \in \mathbb{R}^n \mid f(x) \leq \xi\}.$$

- El conjunto de nivel inferior estricto de f de altura $\xi \in \mathbb{R}$ es:

$$\text{lev}_{< \xi} f = \{x \in \mathbb{R}^n \mid f(x) < \xi\}.$$

Una observación a tener en cuenta es que, aunque estamos considerando a las funciones tomando valores en $\overline{\mathbb{R}}$, hemos definido el *epígrafo*, el *epígrafo estricto* y la *gráfica* de f como subconjuntos de $\mathbb{R}^n \times \mathbb{R}$ y no de $\mathbb{R}^n \times \overline{\mathbb{R}}$. Antes de definir a las funciones convexas, presentemos el siguiente lema.

Lema A.2. Sea $f : \mathbb{R}^n \rightarrow \overline{\mathbb{R}}$ una función. Tenemos $\text{epi}(f)$ es convexo si y solamente si $\text{epi}_s(f)$ es convexo.

La demostración del lema es inmediato, pasemos entonces a definir a las funciones convexas.

Definición A.13 (Función Convexa). Sea $f : \mathbb{R}^n \rightarrow \overline{\mathbb{R}}$ una función. Decimos que f es una **función convexa** si su epígrafo es convexo.

Notemos rápidamente que gracias al lema A.2 es posible dar una definición equivalente de función convexa utilizando el epígrafo estricto, más aun; si $(x, f(x))$ y $(y, f(y))$ pertenecen a $\text{epi}(f)$ y si f es convexo entonces para todo $\alpha \in [0, 1]$ tendremos

$$(\alpha x + (1 - \alpha)y, \alpha f(x) + (1 - \alpha)f(y)) \in \text{epi}(f)$$

además, $\forall x, y \in \mathbb{R}^n$ y $\forall \alpha \in [0, 1]$ se tiene

$$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y). \quad (\text{A.3})$$

Recíprocamente, si (A.3) es verdad, se deduce fácilmente que $\text{epi}(f)$ es convexo, donde (A.3) puede ser tomado también como la definición de la convexidad de una función f (caso que es más frecuente en la literatura). Veamos a continuación dos definiciones ligadas a las funciones convexas.

Definición A.14. Sea $f : \mathbb{R}^n \rightarrow \overline{\mathbb{R}}$ una función.

- Decimos que f es una **función concava** si $-f$ es convexa.
- Decimos que f es una **función afín** si y solo si esta es a la vez convexa y concava.

Introduzcamos ahora los refinamientos siguientes.

Definición A.15. Una función $f : \mathbb{R}^n \rightarrow \overline{\mathbb{R}}$ es **estrictamente convexa** si ella es convexa y si la desigualdad estricta tiene lugar en la definición A.3 cuando $x \neq y$ y $\alpha \in]0, 1[$.

Definición A.16. Una función $f : \mathbb{R}^n \rightarrow \overline{\mathbb{R}}$ es **fuertemente convexa** (de módulo b) si $\exists b > 0 : \forall x, y \in \mathbb{R}^n$ y $\forall \alpha \in [0, 1]$,

$$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y) - b\alpha(1 - \alpha)\frac{\|x - y\|^2}{2}.$$

Una función *fuertemente convexa* será siempre *estrictamente convexa*, pero el recíproco no es siempre verdad. Si bien la función f que estamos estudiando toma valores en $\overline{\mathbb{R}}$, nosotros estamos interesados en los conjuntos donde f toma valores finitos. veamos entonces la siguiente definición

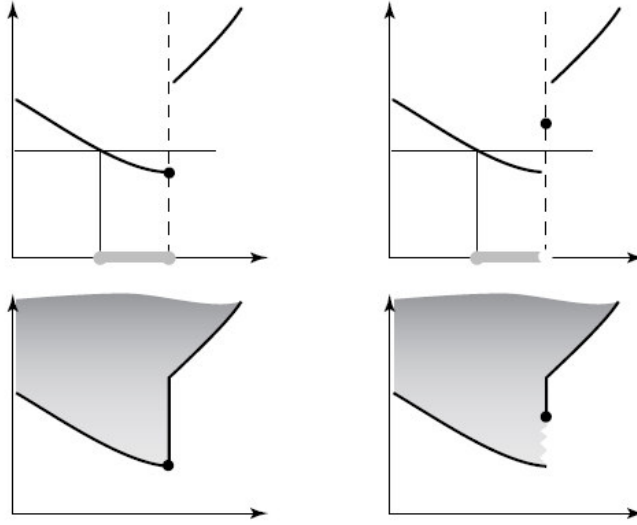


Figura A.3: Una función s.c.i., una modificación no s.c.i., los conjuntos de nivel y sus epígrafos asociados.

Definición A.17. Sea $f : \mathbb{R}^n \rightarrow \overline{\mathbb{R}}$ una función, el **dominio** de f esta dada por:

$$\text{dom}(f) = \{x \in \mathbb{R}^n \mid f(x) < +\infty\}.$$

Proposición A.1. Sea $f : \mathbb{R}^n \rightarrow \overline{\mathbb{R}}$ una función convexa. Entonces, su dominio $\text{dom}(f)$ es un conjunto convexo.

Demostración. Ver [Heinz H. Bauschke, 2017]. □

La teoría de las funciones convexas es más poderosa en presencia de la semicontinuidad inferior. Una propiedad clave de las funciones convexas semicontinuas inferiormente es que a estas se les puede asociar un *minorante afín continuo*.

Definición A.18. Decimos que una función es **semicontinua inferiormente (s.c.i.)** en un punto x , si

$$f(x) \leq \liminf_{y \rightarrow x} f(y).$$

La s.c.i. nos dice que la función puede ser discontinua en x , pero solo se permiten discontinuidades “hacia abajo”. El gráfico A.3 ilustra geoméricamente el concepto de semicontinuidad inferior, vemos en el gráfico izquierdo una función s.c.i. (no convexa) y (a la derecha) una modificación de esta función que ya no es más s.c.i.. Se observa claramente que los conjuntos de nivel (en gris sobre el eje horizontal) son cerrados para la versión s.c.i. pero no lo son para su modificación el cuál tampoco es s.c.i.. Para este último, se observa además que el mínimo de la función no puede ser alcanzado, lo cuál supone la importancia de la s.c.i. para la teoría de la optimización. Análogamente, el epígrafo es cerrado para la función s.c.i. mientras que para su versión no s.c.i. no necesariamente lo es. Estas observaciones son introducidas formalmente en el siguiente teorema.

Teorema A.4. Para una función $f : \mathbb{R}^n \rightarrow \overline{\mathbb{R}}$ (no necesariamente convexa), las siguientes proposiciones son equivalentes:

- f es s.c.i.
- su epígrafo es cerrado.

- sus conjuntos de nivel son cerrados.

Demostración. Ver [Cohen, 2000]. \square

Definición A.19. El conjunto de las funciones convexas y s.c.i. definidas de \mathbb{R}^n en $\overline{\mathbb{R}}$ es denotado por $\Gamma(\mathbb{R}^n)$, es decir:

$$\Gamma(\mathbb{R}^n) \stackrel{def}{=} \{f : \mathbb{R}^n \rightarrow \overline{\mathbb{R}} / f \text{ es convexa y s.c.i.}\}.$$

Si bien, el conjunto $\Gamma(\mathbb{R}^n)$ es cerrado bajo muchas operaciones importantes como por ejemplo, bajo la multiplicación de números reales estrictamente positivos, sin embargo, este posee debilidades en las funciones f tales que $-\infty \in f(\mathbb{R}^n)$ como lo demuestra la siguiente proposición.

Proposición A.2. Sea $f \in \Gamma(\mathbb{R}^n)$ y suponga que $-\infty \in f(\mathbb{R}^n)$. Entonces f no tiene valor real en ninguna parte, es decir, $f(\mathbb{R}^n) \subseteq \{-\infty, +\infty\}$.

Demostración. Ver [Heinz H. Bauschke, 2017]. \square

Esta proposición muestra que las funciones convexas y s.c.i. no propias son de uso limitado. Por el contrario, las funciones propias, convexas y s.c.i. son las mas usadas y es en este tipo de funciones sobre el cuál estamos interesados. Esto motiva las siguientes definiciones.

Definición A.20. Decimos que una función $f : \mathbb{R}^n \rightarrow \overline{\mathbb{R}}$ es **propia** si $-\infty \notin f(\mathbb{R}^n)$ y $\text{dom}(f) \neq \emptyset$.

Definición A.21. El conjunto de las funciones propias, s.c.i y convexas de \mathbb{R}^n en $\overline{\mathbb{R}}$ es denotado por $\Gamma_0(\mathbb{R}^n)$.

Como mencionamos lineas arriba, una propiedad clave de las funciones en $\Gamma_0(\mathbb{R}^n)$ es que cada una de ellas posee un minorante afin continuo. Para ver esto, requerimos la siguiente definición:

Definición A.22 (Continuous affine minorant). Un “**minorante afin continuo**” de una función $f : \mathbb{R}^n \rightarrow \overline{\mathbb{R}}$ esta definido por,

- un “coeficiente director” $m \in \mathbb{R}^n$, y
- una “ordenada en el origen” $\alpha \in \mathbb{R}$,

tales que $\forall y \in \mathbb{R}^n$ se cumpla

$$f(y) \geq \langle m, y \rangle + \alpha, \tag{A.4}$$

donde se dice que este minorante es “exacto” en $x \in \text{dom}(f)$ si la igualdad en (A.4) tiene lugar cuando $y = x$, del cual obtenemos $\alpha = f(x) - \langle m, x \rangle$ y por lo tanto, la desigualdad (A.4) podemos reescribirlo como

$$f(y) \geq \langle m, y - x \rangle + f(x) \tag{A.5}$$

Teorema A.5. Sea $f \in \Gamma_0(\mathbb{R}^n)$. Entonces, f posee un minorante afin continuo. Más precisamente,

$$(\forall y \in \mathbb{R}^n)(\exists x \in \text{dom}(f))(\exists m \in \mathbb{R}^n) / f(y) \geq \langle m, y - x \rangle + f(x) \tag{A.6}$$

Demostración. Ver [Heinz H. Bauschke, 2017]. \square

Corolario A.3. Sea $f \in \Gamma_0(\mathbb{R}^n)$. Entonces, f es acotado inferiormente sobre cada subconjunto no vacío y acotado de \mathbb{R}^n .

Capítulo B

Introducción a los Procesos Estocásticos

En esta sección, el objetivo será representar mediante un modelo matemático el estado de un sistema en función de un *parámetro* y el *azar*. Por lo tanto, si hacemos intervenir el “azar” en la forma axiomatizada habitual de un espacio probabilístico $(\Omega, \mathcal{F}, \mathbb{P})$, y si t denota dicho *parámetro*, entonces el modelo matemático buscado se presenta tan naturalmente como una función

$$(\omega, t) \rightsquigarrow X(\omega, t)$$

definido en el espacio $\Omega \times \mathbb{T}$ y a valores sobre un espacio E el cuál describe los estados posibles del sistema. Esto motiva la siguiente definición:

Definición B.1. Sea $(\Omega, \mathcal{F}, \mathbb{P})$ un espacio de probabilidad, \mathbb{T} un conjunto no vacío, y (E, \mathcal{E}) un espacio medible. Un **proceso estocástico** definido sobre el espacio $(\Omega, \mathcal{F}, \mathbb{P})$ y con valores en (E, \mathcal{E}) es una función X definida sobre el espacio $\Omega \times \mathbb{T}$ que toma valores en E , tal que para cada $t \in \mathbb{T}$, la función

$$\omega \rightsquigarrow X_t(\omega)$$

sea medible de (Ω, \mathcal{F}) en (E, \mathcal{E})

El conjunto \mathbb{T} será llamado el *conjunto de parámetros* del proceso estocástico y al nivel interpretativo, este puede ser en general por ejemplo el tiempo, posiciones espaciales o espacio-tiempo, etc. En algunos casos será necesario suponer que con \mathbb{T} viene asociado un espacio medible $(\mathbb{T}, \mathcal{T})$ y en otros casos un espacio topológico $(\mathbb{T}, \mathcal{B}(\mathbb{T}))$, en general, \mathbb{T} es del todo arbitrario. Todo esto hasta el momento en que introduzcamos *las filtraciones*; con la introducción de este concepto, restringiremos la atención al caso cuando $\mathbb{T} \subseteq [0, \infty)$, $\mathcal{T} = \mathcal{B}(\mathbb{T})$ e interpretaremos a \mathbb{T} como el conjuntos de los tiempos. Todo el desarrollo avanzado de la tesis tendrá lugar cuando $\mathbb{T} = [0, \infty)$ (o $\mathbb{T} = [0, t_0]$), sin embargo, es conceptualmente interesante observar estos primeros párrafos en una forma más general para el cual supondremos solo que \mathbb{T} es un conjunto no vacío. El espacio $(\Omega, \mathcal{F}, \mathbb{P})$ es llamado a veces el *espacio de probabilidad subyacente* del proceso, mientras $(E, \mathcal{B}(E))$ es el *espacio de estados* o *espacio fase*.

Si analizamos detenidamente la definición B.1, este presenta numerosas observaciones:

Observación:

- (a) Del mismo modo que la noción de variable aleatoria era relativo a una estructura de espacio medible (Ω, \mathcal{A}) y no a un espacio medido $(\Omega, \mathcal{A}, \mathbb{P})$, la noción de proceso estocástico no exige verdaderamente definir una ley de probabilidad \mathbb{P} , esta ley será introducida más adelante cuando ya tengamos claro el espacio donde se desarrollan dichos procesos estocásticos.
- (b) Nosotros hemos definido un proceso estocástico como una función $X : \Omega \times \mathbb{T} \rightarrow E$, sin embargo, este también puede ser visto desde dos puntos de vista complementarios:
- Como una aplicación que a cada $t > 0$ fijo asocia una v.a $X(\omega, t) = X_t(\omega)$, el cuál describe el *estado* del sistema en el instante t ; en este caso el proceso estocástico es una familia $(X_t)_{t \in \mathbb{T}}$ de v.a., y en consecuencia como una aplicación de \mathbb{T} en el conjunto de variables aleatorias con valores en E .
 - Como una aplicación que a cada $\omega \in \Omega$ fijo asocia una entera *trayectoria* $X(\omega, t) = X_\omega(t)$ del proceso (una aplicación $\omega \mapsto (t \mapsto X_\omega(t))$); en este segundo caso el proceso estocástico consiste en el conjunto(pesado) de todas las trayectorias posibles del proceso. Debemos advertir que en esta última interpretación, los procesos estocásticos se presentan como una variable aleatoria que toma valores en el conjunto de las trayectorias, es decir como una “*función aleatoria*”; sin embargo, esta noción no es muy correcta desde el punto de vista matemático pues faltaría definir un σ -*algebra* sobre el conjunto de las trayectorias.

Estos dos puntos de vista son sustancialmente equivalentes y vienen adoptados según la necesidad o condiciones del fenómeno. Cualquiera que sea el caso, dado que un proceso estocástico es una función sobre $\Omega \times \mathbb{T}$, es necesario por lo tanto saber bajo que condiciones este conjunto es medible. Esto nos permite dar la siguiente definición:

Definición B.2. *Supongamos que \mathbb{T} esta equipado con un σ -algebra \mathcal{T} . Se dice que el proceso X (indexado por el espacio de medida $(\mathbb{T}, \mathcal{T})$) es MEDIBLE si la aplicación*

$$X : \Omega \times \mathbb{T} \rightarrow E$$

$$X(\omega, t) := X_t(\omega)$$

es $(\mathcal{F} \otimes \mathcal{T}, \mathcal{E})$ -medible.

Sobre el espacio producto $\Omega \times \mathbb{T}$, $\mathcal{F} \otimes \mathcal{T}$ es el σ -*algebra* PRODUCTO, es decir el σ -*algebra* generado por los productos cartesianos del tipo $A \times B$, con $A \in \mathcal{F}$ y $B \in \mathcal{T}$. Se observa claramente que en esta definición, cuando \mathbb{T} es numerable finito, el σ -*algebra* \mathcal{T} puede ser elegido como $2^{\mathbb{T}}$, y aquí, todos los procesos estocásticos serán medibles.

Pero: ¿para que sirve tal medida que acabamos de definir en B.2? Directamente de esta definición se tiene que, para cada subconjunto finito de índices (t_1, \dots, t_n) , la función

$$X_{(t_1, \dots, t_n)} : \Omega \rightarrow E^n$$

$$\omega \mapsto (X_{t_1}(\omega), \dots, X_{t_n}(\omega))$$

es $(\mathcal{F}, \otimes^n \mathcal{E})$ -MEDIBLE. De aquí tenemos la siguiente definición.

Definición B.3 (FDD). Sean $t_1 < \dots < t_n \in \mathbb{T}$, la función

$$\omega \mapsto (X_{t_1}(\omega), \dots, X_{t_n}(\omega))$$

es medible de (Ω, \mathcal{F}) en $(E^n, \otimes^n \mathcal{E})$. Las leyes de estos vectores (leyes imágenes de \mathbb{P} respecto a estas aplicaciones) se llaman **distribuciones de dimensión finita (FDD)** del proceso:

$$\mathbb{P}_{(X_{t_1}, \dots, X_{t_n})} = X_{(t_1, \dots, t_n)}(\mathbb{P})$$

La definición B.3 tiene una interpretación física bastante interesante, esto es, de experimentos o simulaciones numéricas, por lo general solo podemos obtener información sobre las FDDs del proceso. Por lo tanto, es necesario preguntar: ¿Las FDDs de un proceso estocástico son suficientes para determinar un proceso estocástico de manera única?, ¿Bajo que condiciones dos procesos describen el mismo fenómeno? y por otra parte: dado un fenómeno natural, ¿como usaremos las observaciones empíricas para construir un proceso que describa tal fenómeno?

La respuesta a la primera pregunta es afirmativa para procesos con trayectorias continuas, pero, para procesos en general esto no es verdad, volveremos a esto más adelante. La respuesta clásica a las dos últimas preguntas es la siguiente. Para todos los sistemas finitos de instantes t_1, t_2, \dots, t_n , supongamos que es posible determinar el estado del proceso en estos momentos con una precisión arbitraria. Efectuando un gran número de experiencias independientes, podemos entonces evaluar con una precisión arbitraria las probabilidades del tipo

$$\mathbb{P}(X_{t_1} \in A_1, \dots, X_{t_n} \in A_n) \quad \text{con} \quad (A_1, \dots, A_n \in \mathcal{E}).$$

De aquí, tenemos la siguiente definición:

Definición B.4.

- Dos procesos estocásticos $\mathcal{X} = (X_t)_{t \in \mathbb{T}}$ y $\mathcal{Y} = (Y_t)_{t \in \mathbb{T}}$ (definidas no necesariamente sobre el mismo espacio) se dice que son **equivalentes** si tienen las mismas distribuciones de dimensión finita.
- Dos procesos estocásticos $\mathcal{X} = (X_t)_{t \in \mathbb{T}}$ y $\mathcal{Y} = (Y_t)_{t \in \mathbb{T}}$ (definidas sobre el mismo espacio) se dice que son **modificación** (o **versión**) uno del otro, si para cada $t \in \mathbb{T}$ tenemos

$$\mathbb{P}(X_t = Y_t) = 1,$$

es decir, si sus trayectorias coinciden casi seguramente.

- Dos procesos estocásticos $\mathcal{X} = (X_t)_{t \in \mathbb{T}}$ y $\mathcal{Y} = (Y_t)_{t \in \mathbb{T}}$ (definidas sobre el mismo espacio) se dice que son **indistinguibles** si

$$\mathbb{P}(X_t = Y_t \ \forall t \in \mathbb{T}) = 1,$$

es decir, si todas sus trayectorias de los dos procesos coinciden en cada t , con la posible excepción de un subconjunto de medida nula.

Las tres definiciones son bastante diferentes. Por ejemplo, es fácil ver que dos procesos indistinguibles son modificación uno del otro y también son equivalentes, pero el viceversa no es cierto, es decir, dos procesos que son equivalentes no son indistinguibles. En efecto, sean

$$N_t = \overline{\{X_t = Y_t\}}, \quad t > 0$$

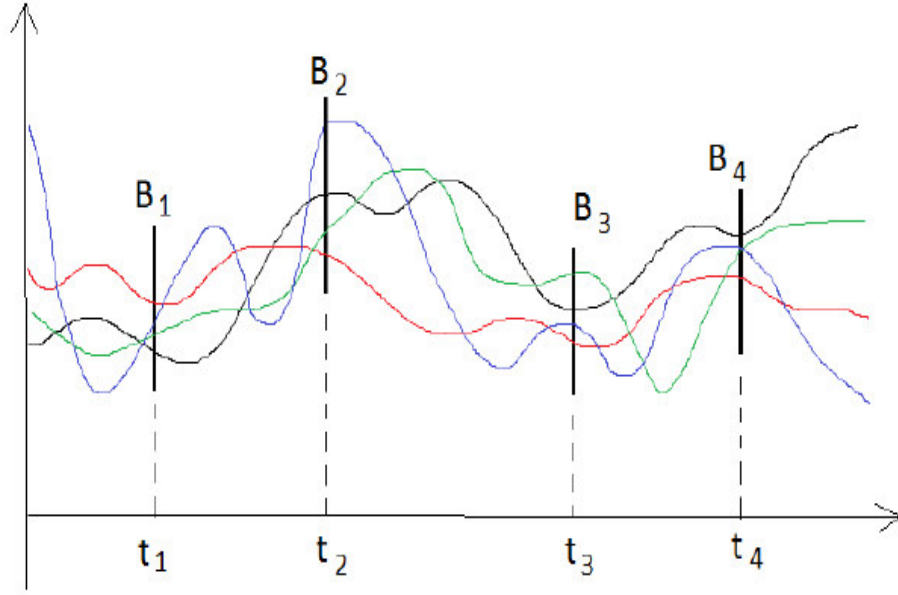


Figura B.1: Imágen intuitiva de un conjunto cilíndrico.

$$N = \overline{\{X_t = Y_t, \forall t > 0\}} = \bigcup N_t$$

la indistinguibilidad requiere que $\mathbb{P}(N) = 0$ y por lo tanto también $\mathbb{P}(N_t) = 0, \forall t > 0$, es decir implica la equivalencia. La simple equivalencia, por el contrario, requiere solo que $\mathbb{P}(N_t) = 0, \forall t > 0$, pero esto no es suficiente para tener que $\mathbb{P}(N) = \mathbb{P}(\bigcup N_t) = 0$ a causa del hecho que el conjunto de los $t > 0$ es no numerable. De manera análoga, dos procesos equivalentes no necesariamente son modificación uno del otro.

B.0.1. Construcción Formal de la Estructura Probabilística de un Proceso Estocástico

Consideremos un proceso estocástico $\mathcal{X} = (X_t)_{t \in \mathbb{T}}$ medible, con

$$X_t : (\Omega, \mathcal{F}) \rightarrow (E, \mathcal{E})$$

con (E, \mathcal{E}) un espacio medible para cada $t \in \mathbb{T}$. Como dijimos anteriormente, este puede ser visto como una variable aleatoria:

$$\begin{aligned} \mathcal{X} : (\Omega, \mathcal{F}, \mathbb{P}) &\rightarrow (E^{\mathbb{T}}, \mathcal{E}^{\otimes \mathbb{T}}) \\ \omega &\rightarrow X(\omega) : \mathbb{T} \rightarrow E \\ &\quad t \rightarrow X_t(\omega) \end{aligned}$$

donde obviamente $E^{\mathbb{T}} = \mathcal{L}(\mathbb{T}, E)$. Para garantizar que \mathcal{X} sea efectivamente una variable aleatoria, debemos asegurarnos que esta sea $(\mathcal{F}, \mathcal{E}^{\otimes \mathbb{T}})$ -MEDIBLE.

Construcción del σ -álgebra $\mathcal{E}^{\otimes \mathbb{T}}$

Habíamos definido el σ -álgebra producto solo en el caso de un número finito de σ -álgebras, mientras que aquí, en general tenemos que $|\mathbb{T}| \geq \aleph_0$.

- Indiquemos con Υ al conjunto de todas las n -*uplas* ordenadas $\tau = (t_1, \dots, t_n)$ de elementos de \mathbb{T} tales que $t_i \neq t_j$ para cada $i \neq j$, con $i, j = 1, \dots, n$.
- Indiquemos con $\mathcal{E}^{\otimes n}$ el σ -álgebra producto de n copias de \mathcal{E} .

Dados un $\tau = (t_1, \dots, t_n) \in \Upsilon$ y un $B \in \mathcal{E}^{\otimes n}$ consideremos el conjunto

$$C(\tau, B) \subset E^{\mathbb{T}}$$

de todas las funciones $f : \mathbb{T} \rightarrow E$ tales que

$$(f(t_1), \dots, f(t_n)) \in B.$$

Este conjunto es conocido con el nombre de **conjunto cilíndrico** (con base B y coordenadas t_1, \dots, t_n). La figura (B.1) muestra una gráfica intuitiva de este conjunto.

Sea \mathcal{A}_0 la familia de todos los conjuntos cilíndricos $C(\tau, B)$, con $n \in \mathbb{N}$, $\tau = (t_1, \dots, t_n) \in \Upsilon$ y $B \in \mathcal{E}^{\otimes n}$. La familia \mathcal{A}_0 es un álgebra de $E^{\mathbb{T}}$. En efecto, sin ahondar mucho en detalles tenemos:

- $E^{\mathbb{T}} \in \mathcal{A}_0$ para esto es suficiente tomar $\tau = (t_1)$ y $B = E$ luego $E^{\mathbb{T}} = C(\tau, B)$.
- El complemento de un conjunto de la forma $\{(f(t_1), \dots, f(t_n)) \in B\}$ es el conjunto cilíndrico $\{(f(t_1), \dots, f(t_n)) \in B^c\}$, luego si $C(\tau, B) \in \mathcal{A}_0$ entonces también $(C(\tau, B))^c \in \mathcal{A}_0$.
- La última propiedad es un poco extensa de escribir en general, pero es obvia si se piensa, por ejemplo, en el caso particular de los dos conjuntos cilíndricos $\{f(t_1) \in B_1\}$ y $\{f(t_2) \in B_2\}$, con $\tau = (t_1, t_2)$ y $B_1, B_2 \in \mathcal{E}$, cuya intersección es el conjunto $\{(f(t_1), f(t_2)) \in B_1 \times B_2\}$ el cuál claramente pertenece a \mathcal{A}_0 .

De aquí, tenemos que $\mathcal{E}^{\otimes \mathbb{T}}$ será el más pequeño σ -álgebra de $E^{\mathbb{T}}$ que contiene a \mathcal{A}_0 . Entre los elementos de $\mathcal{E}^{\otimes \mathbb{T}}$ estan por supuesto, aquellos de las forma:

$$\{f \in E^{\mathbb{T}} : f(t_i) \in B_i, \forall i \in \mathbb{N}\} \quad (\text{B.1})$$

donde $\{t_i\}$ es una sucesión de elementos de \mathbb{T} , $\{B_i\}$ es una sucesión de elementos de \mathcal{E} (son intersecciones numerables de conjuntos cilíndricos).

Ley de un Proceso Estocástico

Dado un proceso estocástico $X = (X_t)_{t \in \mathbb{T}}$ definido sobre un espacio de probabilidad $(\Omega, \mathcal{F}, \mathbb{P})$ y con valores en el espacio medible (E, \mathcal{E}) . Tenemos la siguiente proposición:

Proposición B.1. *La aplicación*

$$\omega \mapsto X.(\omega) \quad (\text{B.2})$$

definida de (Ω, \mathcal{F}) en $(E^{\mathbb{T}}, \mathcal{E}^{\otimes \mathbb{T}})$ es medible, y por lo tanto, $(E^{\mathbb{T}}, \mathcal{E}^{\otimes \mathbb{T}}, \mathbb{P}_X)$ es un espacio de probabilidad el cuál esta caracterizada por sus valores sobre los conjuntos cilíndricos, es decir:

$$\mathbb{P}_X(C(\tau, B)) = \mathbb{P}((X_1, \dots, X_n) \in B)$$

para cada $\tau = (t_1, \dots, t_n) \in \Upsilon$ y $B \in \mathcal{E}^{\otimes n}$.

Demostración. Ver [Baldi, 2001]. □

Definición B.5. *La ley de probabilidad dada en la proposición B.1 es conocida como la **ley del proceso estocástico** X .*

B.0.2. Filtración

Ahora supongamos que \mathbb{T} es un intervalo de \mathbb{R} , más precisamente,

$$\mathbb{T} = [0, \infty).$$

Definición B.6.

- (a) Llamamos **filtración** en $(\Omega, \mathcal{F}, \mathbb{P})$ a una familia $(\mathcal{F}_t)_{t \in \mathbb{T}}$ de σ -álgebras de conjuntos de Ω , $\mathcal{F}_t \subseteq \mathcal{F}$ para cada $t \in \mathbb{T}$, que sea creciente: $\mathcal{F}_s \subset \mathcal{F}_t$ si $s < t \in \mathbb{T}$.
- (b) Un proceso $X = (X_t)_{t \in \mathbb{T}}$ definido sobre un espacio de probabilidad $(\Omega, \mathcal{F}, \mathbb{P})$ y con valores en el espacio medible (E, \mathcal{E}) se dice que es **adaptado** a la filtración $(\mathcal{F}_t)_{t \in \mathbb{T}}$ si para cada $t \in \mathbb{T}$, la función X_t es medible de (Ω, \mathcal{F}_t) en (E, \mathcal{E}) .

Concluimos este capítulo exhibiendo la completación de una filtración. Decimos que un conjunto $N \subset \Omega$ es “insignificante” respecto a $(\Omega, \mathcal{F}, \mathbb{P})$ si $\inf\{\mathbb{P}[B] : B \in \mathcal{F}, N \subset B\} = 0$, es decir, si existe $B \in \mathcal{F}$ tal que $N \subset B$ y $\mathbb{P}[B] = 0$. Indicamos con \mathcal{N} a la clase de todos los conjuntos insignificantes respecto a $(\Omega, \mathcal{F}, \mathbb{P})$. Una filtración $(\mathcal{F}_t)_{t \in \mathbb{T}}$ por lo tanto será *completa* si cada \mathcal{F}_t contiene a \mathcal{N} . Este pequeño detalle técnico en realidad es muy importante ya que de otro modo se crean muchas complicaciones un poco innaturales. De aquí en adelante cuando hablamos de filtración, se entenderá que esta es completa.

Capítulo C

Lista de algoritmos en Matlab

C.1. Algoritmos del Capítulo 3

Algoritmo: ADMM

```
function [z, h] = ADMM_FC_Box(A,b,c,L,U,alpha, QUIET)
% Resuelve el problema de una función cuadrática via ADMM,
% es decir:
%
%   Minimiza 1/2*x'Ax+bx+c;
%   s.a.      C = {x | L <= x <= U}
%
%   o equivalentemente
%
%   Minimiza 1/2*x'Ax+bx+c; + Pro_{C}
%
% INPUT:
%*****
%
% A      : Matriz cuadrada de orden "n"
% b      : vector columna de dimensión "n"
% c      : número real
% alpha  : Parámetro de regularización de la función Box (>0)
% QUIET  : Boolean, si es F (0) se muestra los mensajes T(1) no se
% muestra.
%
% OUTPUT
%*****
%
% x      : Vector solución
% h      : Estructura que contiene:
%         history.objval : Valor del objetivo
% pg_toc : Tiempo empleado

t_start = tic;

MAX_ITER = 1000;
ABSTOL   = 1e-4;
RELTOL   = 1e-2;

[m, n] = size(A);

% Variables
x = zeros(n,1);
z = zeros(n,1);
u = zeros(n,1);

if ~QUIET
    fprintf('%3s\t%10s\t%10s\t%10s\t%10s\t%10s\n', 'iter', ...
        'r norm', 'eps pri', 's norm', 'eps dual', 'objetivo');
end

% cache the factorization
[LA UA] = factor(A, alpha);
f = @(u) 0.5*u'*A*u+b'*u+c;
%INV = inv(eye(n)+alpha*A);

M = eye(n)+alpha*A;
```

```

for k = 1:MAX_ITER

    % x-update
    %x = INV*(z-u-alpha*b);
    x = UA \ ( LA \ (z-u-alpha*b));

    % z-update with relaxation
    zold = z;
    x_hat = alpha*x + (1 - alpha)*zold;
    z = prox_Box(x_hat + u, L, U);

    % u-update
    u = u + (x_hat - z);

    % diagnostics, reporting, termination checks
    h.objval(k) = f(z);

    h.r_norm(k) = norm(x - z);
    h.s_norm(k) = norm(-(1/alpha)*(z - zold));
    h.eps_pri(k) = sqrt(n)*ABSTOL + RELTOL*max(norm(x), norm(-z));
    h.eps_dual(k) = sqrt(n)*ABSTOL + RELTOL*norm((1/alpha)*u);

    if ~QUIET
        fprintf('%3d\t%10.4f\t%10.4f\t%10.4f\t%10.4f\t%10.2f\n', k,
...
            h.r_norm(k), h.eps_pri(k), ...
            h.s_norm(k), h.eps_dual(k), h.objval(k));
    end

    if (h.r_norm(k) < h.eps_pri(k) && ...
        h.s_norm(k) < h.eps_dual(k))
        break;
    end

end
toc(t_start);
end

function x = prox_Box(v, L, U)
% PROX_L1 The proximal operator of the l1 norm.
%
% prox_l1(v,lambda) is the proximal operator of the l1 norm
% with parameter lambda.

    x = max(L, min(v, U));
end

function [L U] = factor(A, alpha)
    n = length(A);
    L = chol( speye(n) + alpha*A, 'lower' );
    U = L';
    %L = sparse(L);
    %U = sparse(L');
end

```


Algoritmo: Gradiente Proximal

```
function [x, history] = PG_FC_Box(A,b,c,l,u,alpha, beta, QUIET)

% Resuelve el problema de una función cuadrática via Gradiente
% Proximal,
% es decir:
%
%   Minimiza  $1/2*x'Ax+bx+c$ ;
%   s.a.
%        $C = \{x \mid l \leq x \leq u\}$ 
%
%   o equivalentemente
%
%   Minimiza  $1/2*x'Ax+bx+c$  +  $Pro_{\{C\}}$ 
%
% INPUT:
%*****
%
% A      : Matriz cuadrada de orden "n"
% b      : vector columna de dimensión "n"
% c      : número real
% alpha  : Parámetro de regularización de la función Box (>0)
% beta   : Parámetro comprendido (0<alpha<1) (default = 1/2)
% QUIET  : Boolean, si es F (0) se muestra los mensajes T(1) no se
% muestra.
%
% OUTPUT
%*****
%
% x      : Vector solución
% history : Estructura que contiene:
%          history.objval : Valor del objetivo
%          pg_toc       : Tiempo empleado

t_start = tic;

MAX_ITER = 1000;
ABSTOL   = 1e-3; %Tolerancia para la función objetivo

f = @(u) 0.5*u'*A*u+b'*u+c;

n = length(A);
M = (0.5)*(A+A');

tic;

%Valor de inicialización
x = zeros(n,1);
xprev = x;
```

```

if ~QUIET
    fprintf('%3s\t%10s\t%10s\t%10s\t%10s\n', 'iter', ...
        'r norm', 'eps pri', 's norm', 'eps dual', 'objetivo');
end

for k = 1:MAX_ITER
    while 1
        grad_f_x = M*x + b;
        z = prox_Box(x - alpha*grad_f_x, l, u);
        if f(z) <= f(x) + grad_f_x'*(z - x) +
            (1/(2*alpha))*sum_square(z - x)
            break;
        end
        alpha = beta*alpha;
    end
    %xprev = x;

    x = z;
    %history.objval(k) = objetivo(A, b, x, x);
    history.objval(k) = f(x);
    %Escribimos el historial
    if ~QUIET
        fprintf('%3d\t%10.4f\n', k, history.objval(k));
    end

    if k > 1 && abs(history.objval(k) - history.objval(k-1)) < ABSTOL
        break;
    end
end

    toc(t_start);

end

function x = prox_Box(v, l, u)
% PROX_Ll    The proximal operator of the l1 norm.
%
%    prox_ll(v,lambda) is the proximal operator of the l1 norm
%    with parameter lambda.

    x = max(l, min(v, u));
end

```

Algoritmo: Gradiente Proximal Acelerado

```
function [x, history] = PGA_FC_Box(A,b,c,l,u,alpha, beta, QUIET)

% Resuelve el problema de una función cuadrática via Gradiente
Proximal,
% es decir:
%
%   Minimiza  $1/2*x'Ax+bx+c$ ;
%   s.a.
%        $C = \{x \mid l \leq x \leq u\}$ 
%
%   o equivalentemente
%
%   Minimiza  $1/2*x'Ax+bx+c$  + Pro_ $\{C\}$ 
%
% INPUT:
%*****
%
% A      : Matriz cuadrada de orden "n"
% b      : vector columna de dimensión "n"
% c      : número real
% alpha  : Parámetro de regularización de la función Box (>0)
% beta   : Parámetro comprendido (0<alpha<1) (default = 1/2)
% QUIET  : Boolean, si es F (0) se muestra los mensajes T(1) no se
muestra.
%
%OUTPUT
%*****
%
% x       : Vector solución
% history : Estructura que contiene:
%           history.objval : Valor del objetivo
% pg_toc  : Tiempo empleado

t_start = tic;

MAX_ITER = 1000;
ABSTOL   = 1e-3; %Tolerancia para la función objetivo

f = @(u) 0.5*u'*A*u+b'*u+c;

n = length(A);
M = (0.5)*(A+A');

tic;

%Valor de inicialización
x = zeros(n,1);
xprev = x;
```

```

if ~QUIET
    fprintf('%3s\t%10s\t%10s\t%10s\t%10s\n', 'iter', ...
        'r norm', 'eps pri', 's norm', 'eps dual', 'objetivo');
end

for k = 1:MAX_ITER
    y = x + (k/(k+3))*(x - xprev);
    while 1
        grad_f_y = M*y + b;
        z = prox_Box(y - alpha*grad_f_y, l, u);
        if f(z) <= f(y) + grad_f_y'*(z - y) +
(1/(2*alpha))*sum_square(z - y)
            break;
        end
        alpha = beta*alpha;
    end
    xprev = x;
    x = z;
    %history.objval(k) = objetivo(A, b, x, x);
    history.objval(k) = f(x);
    %Escribimos el historial
    if ~QUIET
        fprintf('%3d\t%10.4f\n', k, history.objval(k));
    end

    if k > 1 && abs(history.objval(k) - history.objval(k-1)) < ABSTOL
        break;
    end
end

    toc(t_start);

end

function x = prox_Box(v, l, u)
% PROX_L1    The proximal operator of the l1 norm.
%
% prox_l1(v,lambda) is the proximal operator of the l1 norm
% with parameter lambda.

    x = max(l, min(v, u));
end

```

Algoritmo: CVX

```
function [x,cvx_optval]=CVX_FC_Box(f,n,L,U)

%*****
%CVX

t_start = tic;
% n=length(A);
cvx_begin quiet
    cvx_precision low
    variable x(n)
    minimize(f(x))
    subject to
        subject t
            x <= U;
            x >= L;

cvx_end

toc(t_start);
```


C.2. Algoritmos del Capítulo 5

C.2.1. Algoritmos del Primer modelo

Algoritmo: ADMM

```
function [x, h] = ADMM_split_tree(A,b,c,L,U,alpha,QUIET)

% Resuelve el problema de una función cuadrática via Gradiente
% Proximal,
% es decir:
%
%   Minimiza 1/2*x'Ax+bx+c;
%   s.a.
%       C = {x | l <= x <= u}
%
%   o equivalentemente
%
%   Minimiza 1/2*x'Ax+bx+c + Pro_{C}
%
% INPUT:
%*****
%
% A      : Matriz cuadrada de orden "n"
% b      : vector columna de dimensión "n"
% c      : número real
% alpha  : Parámetro de regularización de la función Box (>0)
% beta   : Parámetro comprendido (0<alpha<1) (default = 1/2)
% QUIET  : Boolean, si es F (0) se muestra los mensajes T(1) no se
% muestra.
%
% OUTPUT
%*****
%
% x      : Vector solución
% history : Estructura que contiene:
%          history.objval : Valor del objetivo
%          pg_toc        : Tiempo empleado
addpath('C:\Users\HP\Desktop\tesi\Tesis Real\A-
latex\exp_numerica\Experimentaciones\Formas_Cuadraticas');
%tic;
```

```

MAX_ITER = 100;
ABSTOL   = 1e-4;
RELTOL   = 1e-2;

%M = (0.5)*(A+A');
f = @(u,A) 0.5*u'*A*u+b'*u+c;

if ~QUIET
    fprintf('%3s\t%10s\t%10s\t%10s\t%10s\t%10s\n', 'iter', ...
        'r norm', 'eps pri', 's norm', 'eps dual', 'objetivo');
end

%Creamos las variables

n = length(b);

x_aaa = zeros(n,4); x_aab = zeros(n,4); x_aba = zeros(n,4); x_abb =
zeros(n,4);
x_baa = zeros(n,4); x_bab = zeros(n,4); x_bba = zeros(n,4); x_bbb =
zeros(n,4);

z_0   = zeros(n,1);
z_a   = zeros(n,1); z_b   = zeros(n,1);
z_aa  = zeros(n,1); z_ab  = zeros(n,1); z_ba  = zeros(n,1); z_bb  =
zeros(n,1);
z_aaa = zeros(n,4); z_aab = zeros(n,4); z_aba = zeros(n,4); z_abb =
zeros(n,4);
z_baa = zeros(n,4); z_bab = zeros(n,4); z_bba = zeros(n,4); z_bbb =
zeros(n,4);
z = [ z_aaa z_aab z_aba z_abb z_baa z_bab z_bba z_bbb ];

u_aaa = zeros(n,4); u_aab = zeros(n,4); u_aba = zeros(n,4); u_abb =
zeros(n,4);
u_baa = zeros(n,4); u_bab = zeros(n,4); u_bba = zeros(n,4); u_bbb =
zeros(n,4);

tstart = tic;
%tx = 0;

for k = 1:MAX_ITER

    % x-update
    x_aaa = prox_f(z_aaa - u_aaa, alpha, A(:, :, 1), b, L, U, f);
    x_aab = prox_f(z_aab - u_aab, alpha, A(:, :, 2), b, L, U, f);
    x_aba = prox_f(z_aba - u_aba, alpha, A(:, :, 3), b, L, U, f);
    x_abb = prox_f(z_abb - u_abb, alpha, A(:, :, 4), b, L, U, f);
    x_baa = prox_f(z_baa - u_baa, alpha, A(:, :, 5), b, L, U, f);
    x_bab = prox_f(z_bab - u_bab, alpha, A(:, :, 6), b, L, U, f);
    x_bba = prox_f(z_bba - u_bba, alpha, A(:, :, 7), b, L, U, f);
    x_bbb = prox_f(z_bbb - u_bbb, alpha, A(:, :, 8), b, L, U, f);

    % z-update

    z_0 = avg(x_aaa(:,1), x_aab(:,1), x_aba(:,1), x_abb(:,1), ...
        x_baa(:,1), x_bab(:,1), x_bba(:,1), x_bbb(:,1));

```

```

z_a = avg(x_aaa(:,2), x_aab(:,2), x_aba(:,2), x_abb(:,2));
z_b = avg(x_baa(:,2), x_bab(:,2), x_bba(:,2), x_bbb(:,2));

z_aa = avg(x_aaa(:,3), x_aab(:,3));
z_ab = avg(x_aba(:,3), x_abb(:,3));
z_ba = avg(x_baa(:,3), x_bab(:,3));
z_bb = avg(x_bba(:,3), x_bbb(:,3));

z_aaa = [z_0 z_a z_aa x_aaa(:,4)];
z_aab = [z_0 z_a z_aa x_aab(:,4)];
z_aba = [z_0 z_a z_ab x_aba(:,4)];
z_abb = [z_0 z_a z_ab x_abb(:,4)];
z_baa = [z_0 z_b z_ba x_baa(:,4)];
z_bab = [z_0 z_b z_ba x_bab(:,4)];
z_bba = [z_0 z_b z_bb x_bba(:,4)];
z_bbb = [z_0 z_b z_bb x_bbb(:,4)];

% u-update

u_aaa = u_aaa + x_aaa - z_aaa;
u_aab = u_aab + x_aab - z_aab;
u_aba = u_aba + x_aba - z_aba;
u_abb = u_abb + x_abb - z_abb;
u_baa = u_baa + x_baa - z_baa;
u_bab = u_bab + x_bab - z_bab;
u_bba = u_bba + x_bba - z_bba;
u_bbb = u_bbb + x_bbb - z_bbb;

% diagnostics, reporting, termination checks

x = [ x_aaa x_aab x_aba x_abb x_baa x_bab x_bba x_bbb ];
zold = z;
z = [ z_aaa z_aab z_aba z_abb z_baa z_bab z_bba z_bbb ];
u = [ u_aaa u_aab u_aba u_abb u_baa u_bab u_bba u_bbb ];

%disp(x);
h.r_norm(k) = norm(x - z, 'fro');
h.s_norm(k) = norm(-(z - zold)/alpha, 'fro');
h.eps_pri(k) = sqrt(numel(x))*ABSTOL + RELTOL*max(norm(x, 'fro'),
norm(-z, 'fro'));
h.eps_dual(k) = sqrt(numel(x))*ABSTOL +
RELTOL*norm(u/alpha, 'fro');

A1 = A(:, 1:n, 1); A2 = A(:, n+1:2*n, 1); A3 = A(:, 2*n+1:3*n, 1); A4 =
A(:, 3*n+1:4*n, 1);
S1 =
f(z_aaa(:, 1), A1) + f(z_aaa(:, 2), A2) + f(z_aaa(:, 3), A3) + f(z_aaa(:, 4), A4);
%S1 = S1/4;
A1 = A(:, 1:n, 2); A2 = A(:, n+1:2*n, 2); A3 = A(:, 2*n+1:3*n, 2); A4 =
A(:, 3*n+1:4*n, 2);
S2 =
f(z_aab(:, 1), A1) + f(z_aab(:, 2), A2) + f(z_aab(:, 3), A3) + f(z_aab(:, 4), A4);
%S2 = S2/4;
A1 = A(:, 1:n, 3); A2 = A(:, n+1:2*n, 3); A3 = A(:, 2*n+1:3*n, 3); A4 =
A(:, 3*n+1:4*n, 3);

```



```

    S3 =
f(z_aba(:,1),A1)+f(z_aba(:,2),A2)+f(z_aba(:,3),A3)+f(z_aba(:,4),A4);
    %S3 = S3/4;
    A1 = A(:,1:n,4); A2 = A(:,n+1:2*n,4); A3 = A(:,2*n+1:3*n,4); A4 =
A(:,3*n+1:4*n,4);
    S4 =
f(z_abb(:,1),A1)+f(z_abb(:,2),A2)+f(z_abb(:,3),A3)+f(z_abb(:,4),A4);
    %S4 = S4/4;
    A1 = A(:,1:n,5); A2 = A(:,n+1:2*n,5); A3 = A(:,2*n+1:3*n,5); A4 =
A(:,3*n+1:4*n,5);
    S5 =
f(z_baa(:,1),A1)+f(z_baa(:,2),A2)+f(z_baa(:,3),A3)+f(z_baa(:,4),A4);
    %S5 = S5/4;
    A1 = A(:,1:n,6); A2 = A(:,n+1:2*n,6); A3 = A(:,2*n+1:3*n,6); A4 =
A(:,3*n+1:4*n,6);
    S6 =
f(z_bab(:,1),A1)+f(z_bab(:,2),A2)+f(z_bab(:,3),A3)+f(z_bab(:,4),A4);
    %S6 = S6/4;
    A1 = A(:,1:n,7); A2 = A(:,n+1:2*n,7); A3 = A(:,2*n+1:3*n,7); A4 =
A(:,3*n+1:4*n,7);
    S7 =
f(z_bba(:,1),A1)+f(z_bba(:,2),A2)+f(z_bba(:,3),A3)+f(z_bba(:,4),A4);
    %S7 = S7/4;
    A1 = A(:,1:n,8); A2 = A(:,n+1:2*n,8); A3 = A(:,2*n+1:3*n,8); A4 =
A(:,3*n+1:4*n,8);
    S8 =
f(z_bbb(:,1),A1)+f(z_bbb(:,2),A2)+f(z_bbb(:,3),A3)+f(z_bbb(:,4),A4);
    %S8 = S8/4;
    h.objval(k) = (1/8)*(S1+S2+S3+S4+S5+S6+S7+S8);

    %if mod(k,10) == 0
        fprintf('%3d\t%10.4f\t%10.4f\t%10.4f\t%10.4f\t%10.4f\n', k,
...
            h.r_norm(k), h.eps_pri(k), h.s_norm(k),
h.eps_dual(k),h.objval(k));
    %end

    if h.r_norm(k) < h.eps_pri(k) && h.s_norm(k) < h.eps_dual(k)
        break;
    end

end

%tx/k;
toc(tstart)

end

function x = prox_f(x,alpha,A,b,L,U,f)
    n = length(b);
    I = eye(n);
    A0 = A(:,1:n);
    A1 = A(:,n+1:2*n);
    A2 = A(:,2*n+1:3*n);
    A3 = A(:,3*n+1:4*n);
    % grad_f0 = (0.5)*(A0+A0')*x(:,1)+b;
    % grad_f1 = (0.5)*(A1+A1')*x(:,2)+b;

```

```

%   grad_f2 = (0.5)*(A2+A2')*x(:,3)+b;
%   grad_f3 = (0.5)*(A3+A3')*x(:,4)+b;
%
%   x(:,1) = x(:,1) - alpha*grad_f0;
%   x(:,1)=(I+alpha*A0)\(x(:,1)-alpha*b);
%   x(:,1) = max(L, min(x(:,1), U));
%   x(:,2) = x(:,2) - alpha*grad_f1;
%   x(:,2)=(I+alpha*A1)\(x(:,2)-alpha*b);
%   x(:,2) = max(L, min(x(:,2), U));
%   x(:,3) = x(:,3) - alpha*grad_f2;
%   x(:,3)=(I+alpha*A2)\(x(:,3)-alpha*b);
%   x(:,3) = max(L, min(x(:,3), U));
%   x(:,4) = x(:,4) - alpha*grad_f3;
%   x(:,4)=(I+alpha*A3)\(x(:,4)-alpha*b);
%   x(:,4) = max(L, min(x(:,4), U));

end

function x = avg(varargin)
    N = length(varargin);
    x = 0;
    for k = 1:N
        x = x + varargin[k];
    end
    x = x/N;
end

```

Algoritmo: CVX (solución real)

```

function [h]=CVX_split_tree(A,b,c,L,U)

f = @(u,A) 0.5*u'*A*u+b'*u+c;

n = length(b);
%CVX solution

tic
cvx_begin quiet
    variables y_aaa(n,4) y_aab(n,4) y_aba(n,4) y_abb(n,4) ...
               y_baa(n,4) y_bab(n,4) y_bba(n,4) y_bbb(n,4);
    obj = f(y_aaa(:,1),A(:,1:n,1)) + f(y_aaa(:,2),A(:,n+1:2*n,1)) +
          f(y_aaa(:,3),A(:,2*n+1:3*n,1)) + f(y_aaa(:,4),A(:,3*n+1:4*n,1)) + ...
          f(y_aab(:,1),A(:,1:n,2)) + f(y_aab(:,2),A(:,n+1:2*n,2)) +
          f(y_aab(:,3),A(:,2*n+1:3*n,2)) + f(y_aab(:,4),A(:,3*n+1:4*n,2)) +...
          f(y_aba(:,1),A(:,1:n,3)) + f(y_aba(:,2),A(:,n+1:2*n,3)) +
          f(y_aba(:,3),A(:,2*n+1:3*n,3)) + f(y_aba(:,4),A(:,3*n+1:4*n,3)) +...
          f(y_abb(:,1),A(:,1:n,4)) + f(y_abb(:,2),A(:,n+1:2*n,4)) +
          f(y_abb(:,3),A(:,2*n+1:3*n,4)) + f(y_abb(:,4),A(:,3*n+1:4*n,4)) + ...
          f(y_baa(:,1),A(:,1:n,5)) + f(y_baa(:,2),A(:,n+1:2*n,5)) +
          f(y_baa(:,3),A(:,2*n+1:3*n,5)) + f(y_baa(:,4),A(:,3*n+1:4*n,5)) +...
          f(y_bab(:,1),A(:,1:n,6)) + f(y_bab(:,2),A(:,n+1:2*n,6)) +
          f(y_bab(:,3),A(:,2*n+1:3*n,6)) + f(y_bab(:,4),A(:,3*n+1:4*n,6)) +...
          f(y_bba(:,1),A(:,1:n,7)) + f(y_bba(:,2),A(:,n+1:2*n,7)) +
          f(y_bba(:,3),A(:,2*n+1:3*n,7)) + f(y_bba(:,4),A(:,3*n+1:4*n,7)) +...
          f(y_bbb(:,1),A(:,1:n,8)) + f(y_bbb(:,2),A(:,n+1:2*n,8)) +
          f(y_bbb(:,3),A(:,2*n+1:3*n,8)) + f(y_bbb(:,4),A(:,3*n+1:4*n,8));
    minimize(obj);
    subject to

```

```

% all initial actions must agree
y_aaa(:,1) == y_aab(:,1);
y_aab(:,1) == y_aba(:,1);
y_aba(:,1) == y_abb(:,1);
y_abb(:,1) == y_baa(:,1);
y_baa(:,1) == y_bab(:,1);
y_bab(:,1) == y_bba(:,1);
y_bba(:,1) == y_bbb(:,1);

y_aaa(:,2) == y_aab(:,2);
y_aab(:,2) == y_aba(:,2);
y_aba(:,2) == y_abb(:,2);

y_baa(:,2) == y_bab(:,2);
y_bab(:,2) == y_bba(:,2);
y_bba(:,2) == y_bbb(:,2);

y_aaa(:,3) == y_aab(:,3);
y_aba(:,3) == y_abb(:,3);
y_baa(:,3) == y_bab(:,3);
y_bba(:,3) == y_bbb(:,3);

y_aaa <= U;
y_aab <= U;
y_aba <= U;
y_abb <= U;
y_baa <= U;
y_bab <= U;
y_bba <= U;
y_bbb <= U;

y_aaa >= L;
y_aab >= L;
y_aba >= L;
y_abb >= L;
y_baa >= L;
y_bab >= L;
y_bba >= L;
y_bbb >= L;
cvx_end
toc
disp(y_aab);
disp(y_bbb);
h.p_cvx = cvx_optval/8;
end

```


Algoritmo: CVX (solución VE)

```

function [h]=CVX_split_tree_esp(A,b,c,L,U)

%*****

f = @(u,A) 0.5*u'*A*u+b'*u+c;

n = length(b);
%CVX solution

tic
cvx_begin quiet
    variables w_aaa(n,4) w_aab(n,4) w_aba(n,4) w_abb(n,4) ...
               w_baa(n,4) w_bab(n,4) w_bba(n,4) w_bbb(n,4);
    obj = f(w_aaa(:,1),A(:,1:n)) + f(w_aaa(:,2),A(:,n+1:2*n)) +
f(w_aaa(:,3),A(:,2*n+1:3*n)) + f(w_aaa(:,4),A(:,3*n+1:4*n)) + ...
f(w_aab(:,1),A(:,1:n)) + f(w_aab(:,2),A(:,n+1:2*n)) +
f(w_aab(:,3),A(:,2*n+1:3*n)) + f(w_aab(:,4),A(:,3*n+1:4*n)) +...
f(w_aba(:,1),A(:,1:n)) + f(w_aba(:,2),A(:,n+1:2*n)) +
f(w_aba(:,3),A(:,2*n+1:3*n)) + f(w_aba(:,4),A(:,3*n+1:4*n)) +...
f(w_abb(:,1),A(:,1:n)) + f(w_abb(:,2),A(:,n+1:2*n)) +
f(w_abb(:,3),A(:,2*n+1:3*n)) + f(w_abb(:,4),A(:,3*n+1:4*n)) + ...
f(w_baa(:,1),A(:,1:n)) + f(w_baa(:,2),A(:,n+1:2*n)) +
f(w_baa(:,3),A(:,2*n+1:3*n)) + f(w_baa(:,4),A(:,3*n+1:4*n)) +...
f(w_bab(:,1),A(:,1:n)) + f(w_bab(:,2),A(:,n+1:2*n)) +
f(w_bab(:,3),A(:,2*n+1:3*n)) + f(w_bab(:,4),A(:,3*n+1:4*n)) +...
f(w_bba(:,1),A(:,1:n)) + f(w_bba(:,2),A(:,n+1:2*n)) +
f(w_bba(:,3),A(:,2*n+1:3*n)) + f(w_bba(:,4),A(:,3*n+1:4*n)) +...
f(w_bbb(:,1),A(:,1:n)) + f(w_bbb(:,2),A(:,n+1:2*n)) +
f(w_bbb(:,3),A(:,2*n+1:3*n)) + f(w_bbb(:,4),A(:,3*n+1:4*n));
    minimize(obj);
    subject to
        w_aaa <= U;
        w_aab <= U;
        w_aba <= U;
        w_abb <= U;
        w_baa <= U;
        w_bab <= U;
        w_bba <= U;
        w_bbb <= U;
        w_aaa >= L;
        w_aab >= L;
        w_aba >= L;
        w_abb >= L;
        w_baa >= L;
        w_bab >= L;
        w_bba >= L;
        w_bbb >= L;
cvx_end

h.p_cvx_esp = cvx_optval/8;
toc
end

```

Algoritmo: CVX (solución IP)

```
function [h]=CVX_split_tree_pred(A,b,c,L,U)

%*****

f = @(u,A) 0.5*u'*A*u+b'*u+c;

n = length(b);
%CVX solution

tic
cvx_begin quiet
    variables w_aaa(n,4) w_aab(n,4) w_aba(n,4) w_abb(n,4) ...
               w_baa(n,4) w_bab(n,4) w_bba(n,4) w_bbb(n,4);
    obj = f(w_aaa(:,1),A(:,1:n,1)) + f(w_aaa(:,2),A(:,n+1:2*n,1)) +
f(w_aaa(:,3),A(:,2*n+1:3*n,1)) + f(w_aaa(:,4),A(:,3*n+1:4*n,1)) + ...
f(w_aab(:,1),A(:,1:n,2)) + f(w_aab(:,2),A(:,n+1:2*n,2)) +
f(w_aab(:,3),A(:,2*n+1:3*n,2)) + f(w_aab(:,4),A(:,3*n+1:4*n,2)) +...
f(w_aba(:,1),A(:,1:n,3)) + f(w_aba(:,2),A(:,n+1:2*n,3)) +
f(w_aba(:,3),A(:,2*n+1:3*n,3)) + f(w_aba(:,4),A(:,3*n+1:4*n,3)) +...
f(w_abb(:,1),A(:,1:n,4)) + f(w_abb(:,2),A(:,n+1:2*n,4)) +
f(w_abb(:,3),A(:,2*n+1:3*n,4)) + f(w_abb(:,4),A(:,3*n+1:4*n,4)) + ...
f(w_baa(:,1),A(:,1:n,5)) + f(w_baa(:,2),A(:,n+1:2*n,5)) +
f(w_baa(:,3),A(:,2*n+1:3*n,5)) + f(w_baa(:,4),A(:,3*n+1:4*n,5)) +...
f(w_bab(:,1),A(:,1:n,6)) + f(w_bab(:,2),A(:,n+1:2*n,6)) +
f(w_bab(:,3),A(:,2*n+1:3*n,6)) + f(w_bab(:,4),A(:,3*n+1:4*n,6)) +...
f(w_bba(:,1),A(:,1:n,7)) + f(w_bba(:,2),A(:,n+1:2*n,7)) +
f(w_bba(:,3),A(:,2*n+1:3*n,7)) + f(w_bba(:,4),A(:,3*n+1:4*n,7)) +...
f(w_bbb(:,1),A(:,1:n,8)) + f(w_bbb(:,2),A(:,n+1:2*n,8)) +
f(w_bbb(:,3),A(:,2*n+1:3*n,8)) + f(w_bbb(:,4),A(:,3*n+1:4*n,8));
    minimize(obj);
    subject to
        w_aaa <= U;
        w_aab <= U;
        w_aba <= U;
        w_abb <= U;
        w_baa <= U;
        w_bab <= U;
        w_bba <= U;
        w_bbb <= U;
        w_aaa >= L;
        w_aab >= L;
        w_aba >= L;
        w_abb >= L;
        w_baa >= L;
        w_bab >= L;
        w_bba >= L;
        w_bbb >= L;
cvx_end

h.p_cvx_pres = cvx_optval/8;
toc
end
```

C.2.2. Algoritmos del Segundo modelo

Algoritmo: CVX

```
function
[x,h]=CVX_central(D,Ct,Pmin_t,Pmax_t,Ch,Pmin_h,Pmax_h,Cd,Pmin_d,Pmax_d)

[T,N_ter] = size(Ct);
[T,N_hid] = size(Ch);
[T,N_des] = size(Cd);
Ct = Ct(:);
at = ones(1,N_ter);
Ch = Ch(:);
ah = ones(1,N_hid);
Cd = Cd(:);
ad = ones(1,N_des);
f_t = @(u) Ct'*u(:);
f_h = @(u) sum(Ch);
f_d = @(u) 0.5*u(:)'*u(:);

tic
cvx_begin quiet
    variables x_t(T,N_ter) x_h(T,N_hid) x_d(T,N_des);
    obj = f_t(x_t) + f_h(x_h) + f_d(x_d);
    minimize(obj);
    subject to
        x_t(:,1)+x_t(:,2)+x_t(:,3)+x_t(:,4)+x_t(:,5) + ...
        x_h(:,1)+x_h(:,2)+x_h(:,3) + ...
        x_d(:,1)+x_d(:,2)+x_d(:,3)+x_d(:,4) == D;
        x_t(:,1) <= Pmax_t(1);
        x_t(:,2) <= Pmax_t(2);
        x_t(:,3) <= Pmax_t(3);
        x_t(:,4) <= Pmax_t(4);
        x_t(:,5) <= Pmax_t(5);
        x_h(:,1) <= Pmax_h(1);
        x_h(:,2) <= Pmax_h(2);
        x_h(:,3) <= Pmax_h(3);
        x_d(:,1) <= Pmax_d(1);
        x_d(:,2) <= Pmax_d(2);
        x_d(:,3) <= Pmax_d(3);
        x_d(:,4) <= Pmax_d(4);

        x_t(:,1) >= Pmin_t(1);
        x_t(:,2) >= Pmin_t(2);
        x_t(:,3) >= Pmin_t(3);
        x_t(:,4) >= Pmin_t(4);
        x_t(:,5) >= Pmin_t(5);
        x_h(:,1) >= Pmin_h(1);
        x_h(:,2) >= Pmin_h(2);
        x_h(:,3) >= Pmin_h(3);
        x_d(:,1) >= Pmin_d(1);
        x_d(:,2) >= Pmin_d(2);
        x_d(:,3) >= Pmin_d(3);
        x_d(:,4) >= Pmin_d(4);

cvx_end
toc
x = [x_t x_h x_d];
h.objval = cvx_optval;
end
```


Algoritmo: ADMM

```
function [z, h] =
ADMM_multicentrals(D,Ct,Pmin_t,Pmax_t,Ch,Pmin_h,Pmax_h,Cd,Pmin_d,Pmax_
d,alpha,QUIET)

% Resuelve el problema de una función cuadrática via Gradiente
Proximal,
% es decir:
%
%   Minimiza  $1/2 * x'Ax + bx + c$ ;
%       s.a.
%            $C = \{x \mid l \leq x \leq u\}$ 
%
%   o equivalentemente
%
%   Minimiza  $1/2 * x'Ax + bx + c$  + Pro_ $\{C\}$ 
%
% INPUT:
%*****
%
% A      : Matriz cuadrada de orden "n"
% b      : vector columna de dimensión "n"
% c      : número real
% alpha  : Parámetro de regularización de la función Box (>0)
% beta   : Parámetro comprendido (0<alpha<1) (default = 1/2)
% QUIET  : Boolean, si es F (0) se muestra los mensajes T(1) no se
muestra.
%
% OUTPUT
%*****
%
% x      : Vector solución
% history : Estructura que contiene:
%           history.objval : Valor del objetivo
% pg_toc  : Tiempo empleado
addpath('C:\Users\HP\Desktop\tesi\Tesis Real\A-
latex\exp_numerica\Experimentaciones\Formas_Cuadraticas');

%tic;

MAX_ITER = 10000;
ABSTOL   = 1e-4;
RELTOL   = 1e-2;

[T,N_ter] = size(Ct);
[T,N_hid] = size(Ch);
[T,N_des] = size(Cd);

C = Ct; Ct = Ct(:);
at = ones(1,N_ter);
Ch = Ch(:);
ah = ones(1,N_hid);
Cd = Cd(:);
ad = ones(1,N_des);
%M = (0.5)*(A+A');
```

```

f_t = @(u) Ct'*u(:);
f_h = @(u) sum(Ch);
f_d = @(u) 0.5*u(:)'*u(:);

if ~QUIET
    fprintf('%3s\t%10s\t%10s\t%10s\t%10s\t%10s\n', 'iter', ...
        'r norm', 'eps pri', 's norm', 'eps dual', 'objetivo');
end

%Creamos las variables

z_h = zeros(T,N_hid);
z_t = zeros(T,N_ter);
z_d = zeros(T,N_des);

u_h = zeros(T,N_hid);
u_t = zeros(T,N_ter);
u_d = zeros(T,N_des);

tstart = tic;

z = [z_t z_h z_d];

for k = 1:MAX_ITER
    fprintf('*****\n');
    % x-update

    x_t = z_t - u_t - alpha*C;
    x_h = z_h - u_h;
    x_d = (1/(1+alpha))*(z_d - u_d);

    x = [x_t x_h x_d];
    u = [u_t u_h u_d];
    % z-update
    zold = z;

    z = prox_g_t(x + u, [at ah ad], D, [Pmin_t Pmin_h Pmin_d],
[Pmax_t Pmax_h Pmax_d]);

    % u-update

    u = u + x - z;
    u_t = u(:,1:5); u_h = u(:,6:8); u_d = u(:,9:12);
    z_t = z(:,1:5); z_h = z(:,6:8); z_d = z(:,9:12);

    % diagnostics, reporting, termination checks

    h.r_norm(k) = norm(x(:) - z(:), 'inf');
    h.s_norm(k) = norm(-(z(:) - zold(:))/alpha, 'inf');
    h.eps_pri(k) = sqrt(numel(x))*ABSTOL +
    RELTOL*max(norm(x(:), 'inf'), norm(-z(:), 'inf'));

```



```

h.eps_dual(k) = sqrt(numel(x))*ABSTOL + RELTOL*norm(u(:)/alpha,'inf');

    h.objval(k) = f_t(z_t) + f_h(z_h) + f_d(z_d);

    if ~QUIET
        fprintf('%3d\t%10.4f\t%10.4f\t%10.4f\t%10.4f\t%10.4f\n', k,
        ...
            h.r_norm(k), h.eps_pri(k), h.s_norm(k),
h.eps_dual(k),h.objval(k));
    end

    if h.r_norm(k) < h.eps_pri(k) && h.s_norm(k) < h.eps_dual(k)
        break;
    end

end

%tx/k;
toc(tstart)

end

function x = prox_g_t(x, a, D, Pmin, Pmax)
    [m,n] = size(x);
    aux = [];

    for i = 1:m
        aux = [aux;proj_hyperplane_box(x(i,:),a,D(i,1),Pmin,Pmax)];

    end
    x = aux;

end

```

Algoritmo: Gradiente Proximal

```
function [z, history] =
PG_multicentrals(D,Ct,Pmin_t,Pmax_t,Ch,Pmin_h,Pmax_h,Cd,Pmin_d,Pmax_d,
alpha, beta, QUIET)

% Resuelve el problema lasso via Gradiente Proximal, es decir:
%
%   Minimiza  $1/2 * || Ax + b ||_2^2$ 
%   s.a.
%        $C = \{x \mid l \leq x \leq u\}$ 
%
% o equivalentemente
%
%   Minimiza  $1/2 * || Ax - b ||_2^2 + \text{Pro}_{\{C\}}$ 
%
% INPUT:
%*****
% A      : Matriz m*n
% b      : vector columna de dimensi3n "m"
% alpha  : Par3metro de regularizaci3n de la funci3n Box (>0)
% beta   : Par3metro comprendido (0<beta<1) (default = 1/2)
% QUIET  : Boolean, si es F (0) se muestra los mensajes T(1) no se
muestra.
%
% OUTPUT
%*****
% x      : Vector soluci3n
% history : Estructura que contiene:
%          history.objval : Valor del objetivo
%          pg_toc       : Tiempo empleado

MAX_ITER = 100000;
ABSTOL    = 1e-3; %Tolerancia para la funci3n objetivo

[T,N_ter] = size(Ct);
[T,N_hid] = size(Ch);
[T,N_des] = size(Cd);

%datos para las m3quinas
C = Ct; Ct = Ct(:);
at = ones(1,N_ter);
Ch = Ch(:);
ah = ones(1,N_hid);
Cd = Cd(:);
ad = ones(1,N_des);

%M = (0.5)*(A+A');
f_t = @(u) Ct'*u(:);
f_h = @(u) sum(Ch);
f_d = @(u) 0.5*u(:)'*u(:);

if ~QUIET
    fprintf('%3s\t%10s\t%10s\t%10s\t%10s\t%10s\n', 'iter', ...
        'r norm', 'eps pri', 's norm', 'eps dual', 'objetivo');
end
%Creamos las variables
```

```

x_h = zeros(T,N_hid);
x_t = zeros(T,N_ter);
x_d = zeros(T,N_des);
z_h = zeros(T,N_hid);
z_t = zeros(T,N_ter);
z_d = zeros(T,N_des);
gradf_t = C;
gradf_h = zeros(T,N_hid);

tstart = tic;
for k = 1:MAX_ITER

    while 1
        gradf_d = x_d;
        x = [x_t x_h x_d];
        z = prox_g_t(x - alpha*[gradf_t gradf_h gradf_d], [at ah ad],
D, [Pmin_t Pmin_h Pmin_d], [Pmax_t Pmax_h Pmax_d]);
        z_t = z(:,1:5); z_h = z(:,6:8); z_d = z(:,9:12);

        if (f_t(z_t) + f_h(z_h) + f_d(z_d)) <= ...
            (f_t(x_t) + gradf_t(:)'*(z_t(:) - x_t(:)) +
(1/(2*alpha))*sum_square(z_t(:) - x_t(:))+...
            f_h(x_h) + gradf_h(:)'*(z_h(:) - x_h(:)) +
(1/(2*alpha))*sum_square(z_h(:) - x_h(:))+...
            f_d(x_d) + gradf_d(:)'*(z_d(:) - x_d(:)) +
(1/(2*alpha))*sum_square(z_d(:) - x_d(:)))
            break;
        end
        alpha = beta*alpha;
    end
    %xprev = x;

    x_t = z_t; x_h = z_h; x_d = z_d;

    history.objval(k) = f_t(z_t) + f_h(z_h) + f_d(z_d);
    %Escribimos el historial
    if ~QUIET
        fprintf('%3d\t%10.4f\n', k, history.objval(k));
    end

    if k > 1 && abs(history.objval(k) - history.objval(k-1)) < ABSTOL
        break;
    end
end
toc(tstart)
end

function x = prox_g_t(x, at, D, Pmin, Pmax)
    [m,n] = size(x);
    aux = [];
    for i = 1:m
        aux = [aux;proj_hyperplane_box(x(i,:),at,D(i,1),Pmin,Pmax)];
    end
    x = aux;
end

```

Algoritmo: Gradiente Proximal Acelerado

```
function [z, history] =
PGA_multicentrals(D,Ct,Pmin_t,Pmax_t,Ch,Pmin_h,Pmax_h,Cd,Pmin_d,Pmax_d
,alpha,beta,QUIET)
% Resuelve el problema lasso via Gradiente Proximal, es decir:
%
%   Minimiza  $1/2 * || Ax + b ||_2^2$ 
%   s.a.
%        $C = \{x \mid l \leq x \leq u\}$ 
%
% o equivalentemente
%
%   Minimiza  $1/2 * || Ax - b ||_2^2 + \text{Pro}_{\{C\}}$ 
%
% INPUT:
%*****
% A      : Matriz m*n
% b      : vector columna de dimensi3n "m"
% alpha  : Par3metro de regularizaci3n de la funci3n Box (>0)
% beta   : Par3metro comprendido (0<beta<1) (default = 1/2)
% QUIET  : Boolean, si es F (0) se muestra los mensajes T(1) no se
muestra.
%OUTPUT
%*****
% x      : Vector soluci3n
% history : Estructura que contiene:
%          history.objval : Valor del objetivo
% pg_toc  : Tiempo empleado
MAX_ITER = 100000;
ABSTOL   = 1e-3; %Tolerancia para la funci3n objetivo

[T,N_ter] = size(Ct);
[T,N_hid] = size(Ch);
[T,N_des] = size(Cd);

%datos para las m3quinas t3rmicas
C = Ct; Ct = Ct(:);
at = ones(1,N_ter);
Ch = Ch(:);
ah = ones(1,N_hid);
Cd = Cd(:);
ad = ones(1,N_des);
f_t = @(u) Ct'*u(:);
f_h = @(u) sum(Ch);
f_d = @(u) 0.5*u(:)'*u(:);

if ~QUIET
    fprintf('%3s\t%10s\t%10s\t%10s\t%10s\n', 'iter', ...
        'r norm', 'eps pri', 's norm', 'eps dual', 'objetivo');
end
%Creamos las variables
x_h = zeros(T,N_hid);
x_t = zeros(T,N_ter);
x_d = zeros(T,N_des);
z_h = zeros(T,N_hid);
z_t = zeros(T,N_ter);
z_d = zeros(T,N_des);
```



```

x_prevh = x_h;
x_prevt = x_t;
x_prevd = x_d;
gradfy_t = C;
gradfy_h = zeros(T,N_hid);

tstart = tic;
for k = 1:MAX_ITER
    y_t = x_t + (k/(k+3))*(x_t - x_prevt);
    y_h = x_h + (k/(k+3))*(x_h - x_prevh);
    y_d = x_d + (k/(k+3))*(x_d - x_prevd);
    while 1
        gradfy_d = y_d;
        y = [y_t y_h y_d];
        z = prox_g_t(y - alpha*[gradfy_t gradfy_h gradfy_d], [at ah
ad], D, [Pmin_t Pmin_h Pmin_d], [Pmax_t Pmax_h Pmax_d]);

        z_t = z(:,1:5); z_h = z(:,6:8); z_d = z(:,9:12);

        if (f_t(z_t) + f_h(z_h) + f_d(z_d)) <= ...
            (f_t(y_t) + gradfy_t(:)'*(z_t(:) - y_t(:)) +
(1/(2*alpha))*sum_square(z_t(:) - y_t(:))+...
            f_h(y_h) + gradfy_h(:)'*(z_h(:) - y_h(:)) +
(1/(2*alpha))*sum_square(z_h(:) - y_h(:))+...
            f_d(y_d) + gradfy_d(:)'*(z_d(:) - y_d(:)) +
(1/(2*alpha))*sum_square(z_d(:) - y_d(:)))
            break;
        end
        alpha = beta*alpha;
    end
    x_prevt = x_t;
    x_prevh = x_h;
    x_prevd = x_d;

    x_t = z_t; x_h = z_h; x_d = z_d;
    history.objval(k) = f_t(z_t) + f_h(z_h) + f_d(z_d);
    %Escribimos el historial
    if ~QUIET
        fprintf('%3d\t%10.4f\n', k, history.objval(k));
    end

    if k > 1 && abs(history.objval(k) - history.objval(k-1)) < ABSTOL
        break;
    end
end
toc(tstart)
end

function x = prox_g_t(x, at, D, Pmin, Pmax)
    [m,n] = size(x);
    aux = [];
    for i = 1:m
        aux = [aux;proj_hyperplane_box(x(i,:),at,D(i,1),Pmin,Pmax)];
    end
    x = aux;
end

```

Bibliografía

- Agarwal, Harish, Renaud, John E, Preston, Evan L, & Padmanabhan, Dhanesh. 2004. Uncertainty quantification using evidence theory in multidisciplinary design optimization. *Reliability Engineering & System Safety*, **85**(1-3), 281–294.
- Ahmed, Shabbir, & Sahinidis, Nikolaos V. 1998. Robust process planning under uncertainty. *Industrial & Engineering Chemistry Research*, **37**(5), 1883–1892.
- Almezel, Saleh, Ansari, Qamrul Hasan, & Khamsi, Mohamed Amine. 2014. *Topics in fixed point theory*. Springer.
- Arrow, Kenneth Joseph, & Hurwicz, Leonid. 1958. *Decentralization and computation in resource allocation*. Stanford University, Department of Economics.
- Bai, Dawei, Carpenter, Tamra, & Mulvey, John. 1997. Making a case for robust optimization models. *Management science*, **43**(7), 895–907.
- Baldi, Paolo. 2001. *Equazioni differenziali stocastiche e applicazioni*. Pitagora-Editrice.
- Barbu, Viorel, & Precupanu, Teodor. 2012. *Convexity and optimization in Banach spaces*. Springer Science & Business Media.
- Bauschke, Heinz H, Bolte, Jérôme, & Teboulle, Marc. 2016. A descent lemma beyond Lipschitz gradient continuity: first-order methods revisited and applications. *Mathematics of Operations Research*, **42**(2), 330–348.
- Beale, Evelyn ML. 1955. On minimizing a convex function subject to linear inequalities. *Journal of the Royal Statistical Society. Series B (Methodological)*, 173–184.
- Beck, Amir, & Teboulle, Marc. 2009. Gradient-based algorithms with applications to signal recovery. *Convex optimization in signal processing and communications*, 42–88.
- Bellman, RICHARD. 1957. Dynamic programming. *Princeton, USA: Princeton University Press*, **1**(2), 3.
- Bellman, Richard. 1972. *Dynamic programming*. Princeton University Press.
- Bellman, Richard E, & Zadeh, Lotfi Asker. 1970. Decision-making in a fuzzy environment. *Management science*, **17**(4), B–141.

- Beraldi, P, Musmanno, R, & Triki, C. 1998. *Solving optimal power dispatch via stochastic linear programming with restricted recourse*. Department of Electronics, Informatics and Systems, University of Calabria. Tech. rept. Italy, Technical report.
- Bertsekas, Dimitir P, & Shreve, Steven. 2004. *Stochastic optimal control: the discrete-time case*.
- Billingsley, Patrick. 2012. *Probability and measure*. John Wiley & Sons.
- Billingsley, Patrick. 2013. *Convergence of probability measures*. John Wiley & Sons.
- Birge, John R. 1980. *Solution Methods for Stochastic Dynamic Linear Programs*. Tech. rept. STANFORD UNIV CA SYSTEMS OPTIMIZATION LAB.
- Birge, John R. 1997. State-of-the-art-survey—stochastic programming: computation and applications. *INFORMS journal on computing*, **9**(2), 111–133.
- Birge, John R, & Louveaux, Francois. 2011. *Introduction to stochastic programming*. Springer Science & Business Media.
- Boyd, Stephen, Parikh, Neal, Chu, Eric, Peleato, Borja, Eckstein, Jonathan, *et al.* 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, **3**(1), 1–122.
- Bray, SM, *et al.* 1980. *Control and coordination in hierarchical systems*.
- Breiman, Leo. 1992. *Probability*. SIAM edition.
- Brezis, Haim. 2010. *Functional analysis, Sobolev spaces and partial differential equations*. Springer Science & Business Media.
- Brosilow, CB, Lasdon, LS, & Pearson, JD. 1965. Feasible optimization methods for interconnected systems. *Pages 79–84 of: Joint Automatic Control Conference*.
- Browder, Felix E. 1963a. Nonlinear elliptic boundary value problems. *Bulletin of the American Mathematical Society*, **69**(6), 862–874.
- Browder, Felix E. 1963b. Variational boundary value problems for quasi-linear elliptic equations of arbitrary order. *Proceedings of the National Academy of Sciences*, **50**(1), 31–37.
- Cauchy, Augustin. 1847. Méthode générale pour la résolution des systemes d’équations simultanées. *Comp. Rend. Sci. Paris*, **25**(1847), 536–538.
- Charnes, Abraham, Cooper, William W, & Symonds, Gifford H. 1958. Cost horizons and certainty equivalents: an approach to stochastic programming of heating oil. *Management Science*, **4**(3), 235–263.
- Cohen, Guy. 1984. *Décomposition et coordination en optimisation déterministe différentiable et non différentiable*. Ph.D. thesis.
- Cohen, Guy. 2000. Convexité et optimisation.

- Combettes, Patrick L, & Pesquet, Jean-Christophe. 2007. A Douglas–Rachford splitting approach to nonsmooth convex variational signal recovery. *IEEE Journal of Selected Topics in Signal Processing*, **1**(4), 564–574.
- Combettes, Patrick L, & Pesquet, Jean-Christophe. 2011. Proximal splitting methods in signal processing. *Pages 185–212 of: Fixed-point algorithms for inverse problems in science and engineering*. Springer.
- Combettes, Patrick L, & Van Nguyen, Quang. 2015. Solving composite monotone inclusions in reflexive Banach spaces by constructing best Bregman approximations from their Kuhn-Tucker set. *arXiv preprint arXiv:1505.00362*.
- Combettes, Patrick L, & Wajs, Valérie R. 2005. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling & Simulation*, **4**(4), 1168–1200.
- Conejo, Antonio J, Carrión, Miguel, Morales, Juan M, *et al.* 2010. *Decision making under uncertainty in electricity markets*. Vol. 1. Springer.
- Congreso-Rep-Perú. 2008 (Mayo). “*Promoción de la Inversión para la Generación de Electricidad con el Uso de Energías Renovables*”, *Decreto Legislativo 1002/2008*. Congreso de la República de Perú, Lima-Perú.
- Daldoul, M. 1997. Finite convergence of the partial inverse algorithm. *Journal of optimization theory and applications*, **95**(3), 693–699.
- Dallagi, Anes. 2007. *Méthodes particulières en commande optimale stochastique*. Ph.D. thesis, Université Panthéon-Sorbonne-Paris I.
- Dantzig, George B. 2010. Linear programming under uncertainty. *Pages 1–11 of: Stochastic programming*. Springer.
- Dantzig, George B, & Wolfe, Philip. 1961. The decomposition algorithm for linear programs. *Econometrica: Journal of the Econometric Society*, 767–778.
- Debnath, Lokenath, & Mikusinski, Piotr. 2005. *Introduction to Hilbert spaces with applications*. Academic press.
- Deng, Liang-Jian, Guo, Huiqing, & Huang, Ting-Zhu. 2015. A fast image recovery algorithm based on splitting deblurring and denoising. *Journal of Computational and Applied Mathematics*, **287**, 88–97.
- Deutsch, Frank R. 2012. *Best approximation in inner product spaces*. Springer Science & Business Media.
- Douglas, Jim, & Rachford, Henry H. 1956. On the numerical solution of heat conduction problems in two and three space variables. *Transactions of the American mathematical Society*, **82**(2), 421–439.
- Durrett, Richard. 2018. *Stochastic calculus: a practical introduction*. CRC press.
- Durrett, Rick. 2010. *Probability: theory and examples*. Cambridge university press.

- Dyke, Phil PG. 2012. *An introduction to Laplace transforms and Fourier series*. Springer Science & Business Media.
- Dynkin, EB. 1965. *Markov processes I, II. Grundle. d. math. Wiss.* 121, 122.
- Ekeland, Ivar, & Temam, Roger. 1999. *Convex analysis and variational problems*. Vol. 28. Siam.
- Feller, William. 1968. *An introduction to probability theory and its applications*. Vol. 1. Wiley, New York.
- Feller, William. 2008. *An introduction to probability theory and its applications*. Vol. 2. John Wiley & Sons.
- Gabay, D. 1983. Applications of the method of multipliers to variational inequalities, in *Augmented Lagrange Methods: Applications to the Solution of Boundary Valued Problems*. M. Fortin, R. Glowinski (eds.), North-Holland, Amsterdam, 299–331.
- Gabay, Daniel, & Mercier, Bertrand. 1975. *A dual algorithm for the solution of non linear variational problems via finite element approximation*. Institut de recherche d’informatique et d’automatique.
- Glowinski, Roland, & Marroco, A. 1975. Sur l’approximation, par éléments finis d’ordre un, et la résolution, par pénalisation-dualité d’une classe de problèmes de Dirichlet non linéaires. *Revue française d’automatique, informatique, recherche opérationnelle. Analyse numérique*, **9**(R2), 41–76.
- Golshtein, E. G. 1975. Method for Modification of Monotone Mappings. **11**(6), 1142–1159.
- Heinz H. Bauschke, Patrick L. Combettes. 2017. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer Nature.
- Hestenes, Magnus R. 1969. Multiplier and gradient methods. *Journal of optimization theory and applications*, **4**(5), 303–320.
- Hiriart-Urruty, Jean-Baptiste, & Lemaréchal, Claude. 2013. *Convex analysis and minimization algorithms, Tomes I et II*. Vol. 305. Springer science & business media.
- Horn, Roger A, Horn, Roger A, & Johnson, Charles R. 1990. *Matrix analysis*. Cambridge university press.
- Horner, P. 1999. Planning under uncertainty. Questions & answers with George Dantzig. *OR/MS Today*, **26**, 26–30.
- Infanger, Gerd. 1992. *Planning under uncertainty solving large-scale stochastic linear programs*. Tech. rept. Stanford Univ., CA (United States). Systems Optimization Lab.
- Jacod, Jean, & Protter, Philip. 2004. *Probability essentials*. Springer Science & Business Media.
- Johnson, Norman L, Kemp, Adrienne W, & Kotz, Samuel. 2005. *Univariate discrete distributions*. Vol. 444. John Wiley & Sons.
- Kachurovskii, RI. 1960. Monotone operators and convex functionals. *Uspekhi Matematicheskikh Nauk*, **15**(4), 213–215.

- Kall, Peter, Wallace, Stein W, & Kall, Peter. 1994. *Stochastic programming*. Springer.
- Karimi, Hamed, Nutini, Julie, & Schmidt, Mark. 2016. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. *Pages 795–811 of: Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer.
- Kawata, Tatsuo. 2014. *Fourier analysis in probability theory*. Vol. 15. Academic Press.
- Kellogg, R Bruce. 1969. A nonlinear alternating direction method. *Mathematics of computation*, **23**(105), 23–27.
- Kolmogorov, A. N. 1993. *Grundbegriffe der Wahrscheinlichkeitsrechnung*. Vol. 2. Ergeb. Math.
- Kolmogorov, Andreĭ. *Elements of the theory of functions and functional analysis: AN Kolmogorov and SV Fomin*.
- Lasdon, Leon S. 2002. *Optimization theory for large systems*. Courier Corporation.
- Lasdon, LS, & Schoeffler, James D. 1965. A multi-level technique for optimization. *Pages 85–92 of: Joint Automatic Control Conference*.
- Lee, John. 2010. *Introduction to topological manifolds*. Vol. 202. Springer Science & Business Media.
- Lenoir, Arnaud. 2008. *Modèles et algorithmes pour la planification de production à moyen terme en environnement incertain*. Ph.D. thesis, Université Blaise Pascal-Clermont-Ferrand II.
- Lions, Pierre-Louis, & Mercier, Bertrand. 1979. Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Numerical Analysis*, **16**(6), 964–979.
- Loeve, M. 1978. *Probability Theory I*. Springer-Verlag.
- Malcolm, Scott A, & Zenios, Stavros A. 1994. Robust optimization for power systems capacity expansion under uncertainty. *Journal of the operational research society*, **45**(9), 1040–1049.
- Mann, W Robert. 1953. Mean value methods in iteration. *Proceedings of the American Mathematical Society*, **4**(3), 506–510.
- Marschak, Jacob, & Radner, Roy. 1972. *Economic Theory of Teams*.
- Marschak, Jakob. 1955. Elements for a theory of teams. *Management science*, **1**(2), 127–137.
- Martinet, B. Régularisation d’inéquations variationnelles par approximations successives, *RAIRO* 4 (1970) 154–159. *Google Scholar*.
- Martinet, Bernard. 1972. Détermination approchée d’un point fixe d’une application pseudo-contractante. *CR Acad. Sci. Paris*, **274**(2), 163–165.
- Matsushita, Shin-Ya. 2017. On the convergence rate of the Krasnosel’skiĭ-Mann iteration. *Bulletin of the Australian Mathematical Society*, **96**(1), 162–170.
- MINEM. 2011 (Marzo). “Reglamento de Generación de Electricidad con Energías Renovables”, *Decreto Supremo N° 012-2011-EM*. Ministerio de Energía y Minas, Lima-Perú.

- Minty, George J, *et al.* 1961. On the maximal domain of a “monotone” function. *The Michigan Mathematical Journal*, **8**(2), 135–137.
- Minty, George J, *et al.* 1964. On the monotonicity of the gradient of a convex function. *Pacific Journal of Mathematics*, **14**(1), 243–247.
- Morgenstern, Oskar, & Von Neumann, John. 1953. *Theory of games and economic behavior*. Princeton university press.
- Nadaraya, Elizbar A. 1964. On estimating regression. *Theory of Probability & Its Applications*, **9**(1), 141–142.
- Nesterov, Yurii, *et al.* 2007. *Gradient methods for minimizing composite objective function*.
- Neveu, Jacques. 1965. *Mathematical foundations of the calculus of probability*. Holden-day.
- Oberkampf, William, Helton, Jon, & Sentz, Kari. 2001. Mathematical representation of uncertainty. *Page 1645 of: 19th AIAA Applied Aerodynamics Conference*.
- Osinermin. 2016. *La industria de la electricidad en el Perú: 25 años de aportes al crecimiento económico del país*. Lima-Perú.
- Osinermin. 2018. *Supervisión de contratos de proyectos de Generación y Transmisión de Energía Eléctrica en operación*. Lima-Perú.
- Parzen, Emanuel. 1999. *Stochastic processes*. SIAM.
- Passty, Gregory B. 1979. Ergodic convergence to a zero of the sum of monotone operators in Hilbert space. *J. Math. Anal. Appl.*, **72**, 383–290.
- Peaceman, Donald W, & Rachford, Jr, Henry H. 1955. The numerical solution of parabolic and elliptic differential equations. *Journal of the Society for industrial and Applied Mathematics*, **3**(1), 28–41.
- Picard, Emile. 1890. Mémoire sur la théorie des équations aux dérivées partielles et la méthode des approximations successives. *Journal de Mathématiques pures et appliquées*, **6**, 145–210.
- Polyak, Boris T. 1987. Introduction to optimization. Translations series in mathematics and engineering. *Optimization Software*.
- Powell, Michael JD. 1969. A method for nonlinear constraints in minimization problems. *Optimization*, 283–298.
- Rockafellar, R Tyrell. 1973. The multiplier method of Hestenes and Powell applied to convex programming. *Journal of Optimization Theory and applications*, **12**(6), 555–562.
- Rockafellar, R Tyrrell. 1976a. Augmented Lagrangians and applications of the proximal point algorithm in convex programming. *Mathematics of operations research*, **1**(2), 97–116.
- Rockafellar, R Tyrrell. 1976b. Monotone operators and the proximal point algorithm. *SIAM journal on control and optimization*, **14**(5), 877–898.

- Rockafellar, Ralph. 1966. Characterization of the subdifferentials of convex functions. *Pacific Journal of Mathematics*, **17**(3), 497–510.
- Rockafellar, Ralph Tyrell. 2015. *Convex analysis*. Princeton university press.
- Rockafellar, RT. 1978. Monotone operators and augmented Lagrangian methods in nonlinear programming. *Pages 1–25 of: Nonlinear Programming 3*. Elsevier.
- Rosenthal, Jeffrey S. 2006. *A first look at rigorous probability theory*. World Scientific Publishing Company.
- Rudin, Leonid I, Osher, Stanley, & Fatemi, Emad. 1992. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, **60**(1-4), 259–268.
- Ryu, Ernest K, & Boyd, Stephen. 2016. Primer on monotone operator methods. *Appl. Comput. Math*, **15**(1), 3–43.
- Sahinidis, Nikolaos V. 2004. Optimization under uncertainty: state-of-the-art and opportunities. *Computers & Chemical Engineering*, **28**(6-7), 971–983.
- Schaefer, Helmut H. 1971. Locally Convex Topological Vector Spaces. *Pages 36–72 of: Topological Vector Spaces*. Springer.
- Shapiro, Alexander, Dentcheva, Darinka, & Ruszczyński, Andrzej. 2009. *Lectures on stochastic programming: modeling and theory*. SIAM.
- Shapiro, Jeremy F. 1979. *Mathematical programming: structures and algorithms*. Tech. rept.
- Shiryaev, Albert N. 2016. *Probability-1*. Springer.
- Taylor, S. J. 1973. *Introduction to Measure and Integration*. Cambridge University Press.
- Themelis, Andreas, & Patrinos, Panagiotis. 2016. SuperMann: a superlinearly convergent algorithm for finding fixed points of nonexpansive operators. *arXiv preprint arXiv:1609.06955*.
- Tibshirani, Robert. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 267–288.
- Tintner, Gerhard. 1955. Stochastic linear programming with applications to agricultural economics.
- Tseng, P. 2008. On accelerated proximal gradient methods for convex-concave optimization. *submitted to SIAM Journal on Optimization*, 01.
- Van Nguyen, Quang. 2017. Forward-backward splitting with Bregman distances. *Vietnam Journal of Mathematics*, **45**(3), 519–539.
- Varaiya, P. 1972. Theory of hierarchical, multilevel systems. *IEEE Transactions on Automatic Control*, **17**(2), 280–281.
- Vasudeva, Harkrishan Lal, & Shirali, Satish. 2017. *Elements of Hilbert spaces and operator theory*. Springer.

- von Plato, Jan. 2005. AN Kolmogorov, Grundbegriffe der wahrscheinlichkeitsrechnung (1933). *Pages 960–969 of: Landmark Writings in Western Mathematics 1640-1940*. Elsevier.
- Wallace, Stein W, & Fleten, Stein-Erik. 2003. Stochastic programming models in energy. *Handbooks in operations research and management science*, **10**, 637–677.
- Watson, Eric John. 1981. *Laplace transforms and applications*. Vol. 10. Van Nostrand Reinhold Company.
- Watson, Geoffrey S. 1964. Smooth regression analysis. *Sankhyā: The Indian Journal of Statistics, Series A*, 359–372.
- Wilkinson, JH, & GOLUB, GH. 1966. Note on the Iterative Refinement of Least Squares Solution. *Numerische Mathematik*, **9**, 139–148.
- Wismer, David A. 1971. *Optimization methods for large-scale systems... with applications*. McGraw-Hill Companies.
- Yan, Ming, & Yin, Wotao. 2016. Self equivalence of the alternating direction method of multipliers. *Pages 165–194 of: Splitting Methods in Communication, Imaging, Science, and Engineering*. Springer.